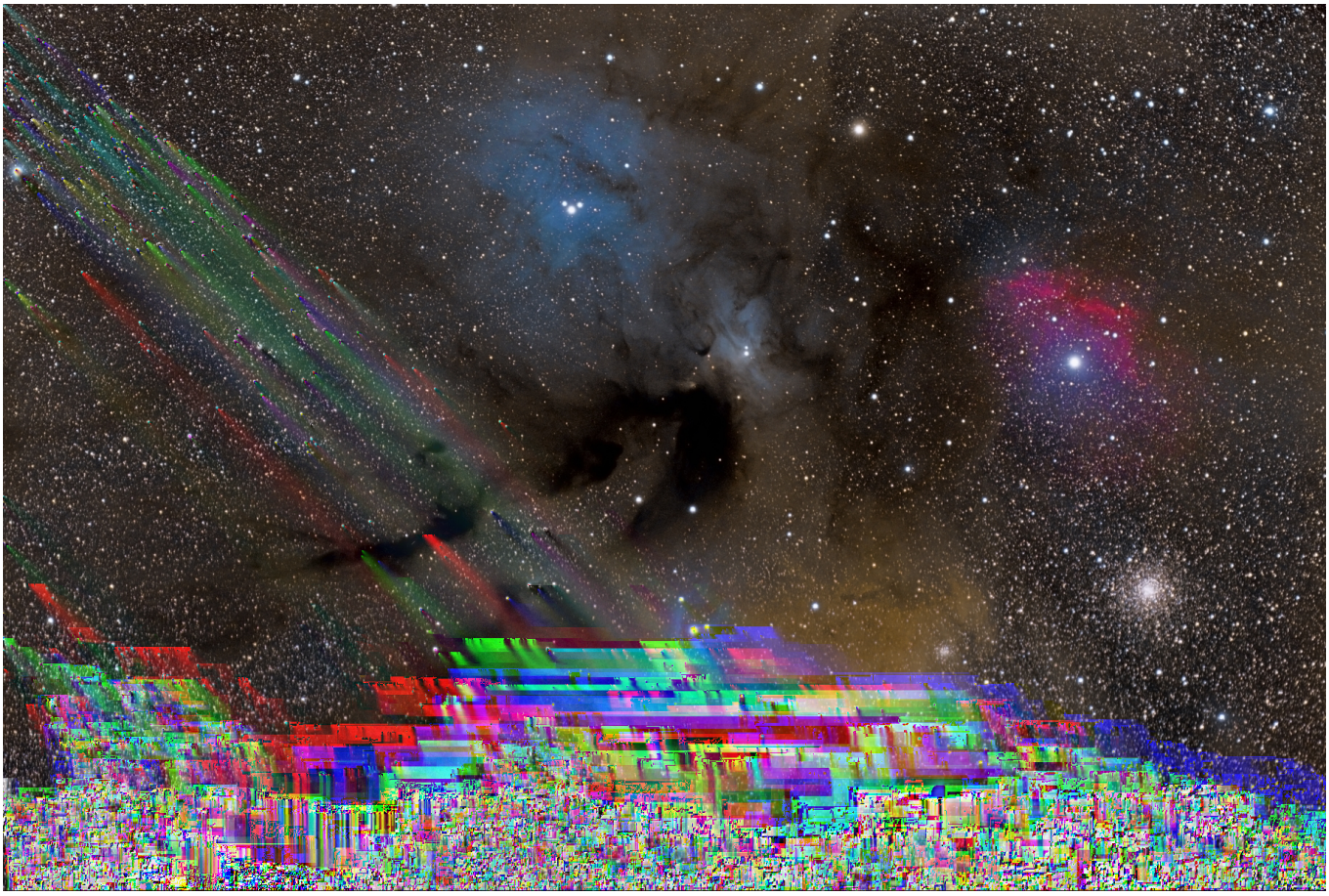
User guide

Siril 0.9.9



Version : 1.53

Date : 06/08/18

Copyright : The free-astro development team, 2018

License : Le contenu est disponible sous licence Public Domain[http://creativecommons.org/licenses/publicdomain/] sauf mention contraire.

Table of contents

Siril 0.9.9 4
1. Siril functions 6
1.1. Siril features 6
1.1.1. **New of 0.9.9 version** 8
1.2. Work on a sequence of converted images 8
1.3. Registration 10
1.3.1. One Star Registration (deep-sky) 13
1.3.2. Global Star Alignment (Deep-sky) 14
1.3.3. Simplified Drizzle x2 17
1.3.4. Image Pattern Alignment (planetary-full disk) 18
1.3.5. Enhanced Correlation Coefficient Maximization (planetary-surfaces) 18
1.3.6. Manual Registration 18
1.4. Plot 19
1.5. Aperture photometry 36
1.6. Stacking 38
1.6.1. Stacking Methods 41
1.6.1.1. Sum Stacking 41
1.6.1.2. Average Stacking With Rejection 41
1.6.1.3. Median Stacking 43
1.6.1.4. Pixel Maximum Stacking 43
1.6.1.5. Pixel Minimum Stacking 43
1.6.2. Normalization 43
1.7. Sequence export 44
1.8. **Available commands** 48
1.9. A powerful astronomy image viewer 73
1.9.1. Images list 78
1.9.2. Colors maps viewer 79
1.9.3. Histograms 84
1.9.4. Right click functions 85
1.9.5. Display use 85
1.10. Image Processing 86
1.10.1. Background extraction 86
1.10.2. Median filter 88
1.10.3. Reducing green tint in processed images 89
1.10.4. Color saturation 90
1.10.5. Banding Reduction 90
1.10.6. Déconvolution 92
1.10.7. RGB compositing... 93
1.10.8. Extraction 95
1.10.8.1. Channel extraction 95
1.10.8.2. Wavelets 97
1.10.9. Cosmetic correction 97
1.10.10. Sequence crop 98
1.11. Analysis 99
1.11.1. Point Spread Function 100

1.11.2. Statistics 107
1.11.3. Noise Estimation 110
1.12. Settings 110
1.12.1. DSLR RAW debayer 110
1.12.1.1. Image Color Ajustement 111
1.12.1.2. Image Matrix Interpolation 111
1.12.1.3. Image Gamma Correction 112
1.12.1.4. Image White Balance 113
1.12.2. Other debayer 113
1.12.3. Camera & Optics 114
1.12.4. Misc. 114
1.13. Contextual help 114
1.14. Keyboard shortcuts 114
2. Tutorials 116
2.1. Deep Sky 116
2.1.1. Convert your images in the FITS format Siril uses (image import) 116
2.1.1.1. Images formats natively supported 117
2.1.1.2. add the files to be converted 117
2.1.2. Pre-processing images 119
2.1.2.1. Prepare dark, flat and bias 120
2.1.2.2. For DSLR images 123
2.1.2.3. Pre-processing and FIT images 124
2.1.3. Special case of demosaicing 126
2.1.4. Use of registration 129
2.1.5. Stacking 132
2.1.6. Process your single images 135
2.1.6.1. Background extraction 136
2.1.6.2. Color calibration tool 145
2.1.6.3. Green Noise Reduction 149
2.1.6.4. Midtones Function Transfer 149
2.1.7. Manual Registration 149
2.1.8. Aperture photometry 154
2.2. Planetary 167
2.2.1. Use SER files 167
2.2.2. Usual planetary processing (avi) 170
2.2.3. Align RGB channels 173
2.2.4. Archiving planetary videos 175
3. Authors 182
Glossary 183
Bibliography 200

Siril 0.9.9

It is an astronomical image processing tool, able to convert, pre-process images, help aligning them automatically or manually, stack them and enhance final images.

M33 by Cyprien Pouzenc

In line with programs like Iris[*http://www.astrosurf.com/buil/iris/iris.htm*] , Siril[p.197] is targeted to **amateur astronomers** having acquired images and wanting to process them in a semi-automatic way.

It provides a more user-friendly interface than Iris' command line as well as more modern and powerful processing algorithms, but it is not yet as automated as DeepSkyStacker or Registax.

It also provides a basic command line to access various processing functions.

Siril is fully capable to pre-process and process **deep sky** images.

One feature still missing for **planetary** images processing is image deformation, but it is still capable of registering and processing them in an elegant way.

The new star detection registration is perfect for deep sky images, but there are also two automatic translation-only registrations, one more suited for planetary and bright nebula images, based on DFT[*https://en.wikipedia.org/wiki/Discrete_Fourier_transform*], and another more simple targeted at deep-sky images, based on PSF[*http://en.wikipedia.org/wiki/Point_spread_function*] on a single reference star.

Additionally, a manual alignment capability makes it also well fitted for amateur astronomers having feature-less or bad quality images that automatic tools fail to align properly.

> Professional astronomers generally use 32-bit depth images, but Siril uses only 16-bit images

internally (except temporary for stacking).

Loading 32-bit images is supported but it will degrade them since they will be converted to 16-bit. It can still be useful if the intent is to produce public-friendly pictures.

And more generally, people who want to use free software on free operating systems to process images. Siril can even be used to extract previewed frames from many videos formats. For an overview, see some image processing results[https://free-astro.org/index.php/Siril:Results].

The current version is 0.9.9

Follow us on Facebook[https://www.facebook.com/Siril-319634734887195/]

Or on Webastro[http://www.webastro.net/forum/showthread.php?t=122969] forum

If you like the software, please help us by contributing with the Donate button here[https://free-astro.org/index.php/Siril]. Siril took us a lot of time and we still have to pay for the server.

1. Siril functions

1.1. Siril features

⚠

New features are being introduced quite regularly.

Native image format support

- Unsigned 16-bit **FITS** files (other FITS are converted to this format on-the-fly)
- 👉SER files
- AVI and many other film files, their support is being dropped in favor of SER.

Image conversion (to the native FITS format only)

- Supported input types: 8-bit and 16-bit **BMP**, **TIFF**, **JPEG**, **PNG** files, **NetPBM** binary images, **RAW** DSLR images.

Pre-processing of images

- With multi-channel offset, dark and flat images.

Image registration; supported methods:

- Global star alignment (rotation + translation)
- Translation using **DFT**[https://en.wikipedia.org/wiki/Cross-correlation] centered on an object, generally used for planetary images
- Translation using **PSF**[https://en.wikipedia.org/wiki/Point_spread_function] of a star, generally used for deep-sky images
- Manual translation with two preview renderings of the current image with reference frame in transparency

💬 Since 0.9.8

- Subpixel alignment for stacking, or more exactly half-pixel alignment, with the feature called simplified drizzle

Registered sequence export

- Supported export file formats: Siril **FITS** sequence, **SER** sequence, uncompressed AVI files, GIF up to 0.9.4, **MP4** and **WEBM** web publishing video formats since 0.9.5
- Optional cropping and resizing of the exported images
- Archiving planetary videos

Image stacking

- Summing
- Median
- Percentile clipping
- Sigma clipping
- Median sigma clipping
- Winsorized sigma clipping
- Linear fit clipping
- Pixel minimum/maximum

Since 0.9.8

- Sum stacking now runs in parallel

16-bit normalization of 8-bit images stacked with deviant pixel rejection algorithms is possible since version 0.9.9

Enhancement and processing of final images

- Lightness/contrast sliders on each layer, different scaling modes (linear, log, square root, squared, asinh, histogram equalization), negative and false color rendering and clipping
- Background removal tool
- Cosmetic correction tool
- Lucy-Ritcharson deconvolution tool

Command line for various processing functions

- Available from the GUI, see the list of available 👉commands.

Each command is documented.

Scripts

Since version 0.9.9 it is possible to create scripts to automate processing.

Photometry features

- A star finding algorithm with **PSF** information
- Light curve data output suitable for plot tools, for one star in a sequence (***working in progress***)

  Today,Siril uses the curves during the registration. So it can be a star, if "one-star registration" is used, or the average of the detected stars.

  For planetary, quality is used (which goes from 0 to 1).

Image compositing tool

- Image compositing tool, combining and aligning multiple layers (1 Luminance + 1 to 7 possible colors) with custom colors into a resulting image, with luminance layer support

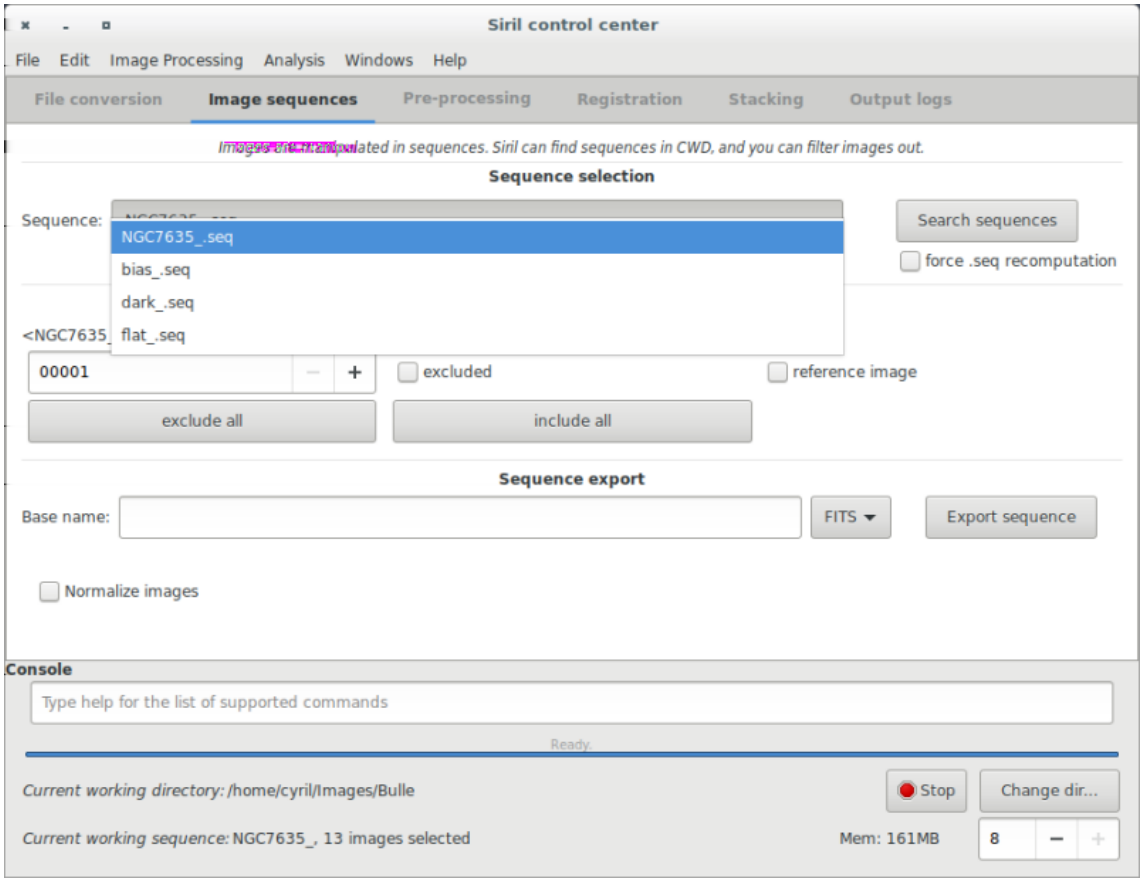
> Leveraging the new scripting, Siril 0.9.9 has been integrated as an automatic way of processing acquisitions in the capture software CCDciel*[https://www.ap-i.net/ccdciel/fr/start]*.

1.1.1. New of 0.9.9 version

- Major update of the command line, with completion and documentation in the GUI, enhanced scripting capability by running commands from a file and also allowing it to be run with no X11 server running with the -s command line option.
- Added commands to stack and register a sequence
- Image statistics, including auto-stretch parameters and stacking normalization, are now cached in the seq file for better performance
- Global star registration now runs in parallel
- Workflow improvement by adding demosaicing as last part of the preprocessing
- Added a filtering method for stacking based on star roundness
- Added an option to normalize stacked images to 16-bit with average rejection algorithm
- All GUI save functions are now done in another thread
- Improved histogram transformation GUI
- Improved support of various FITS pixel formats
- Added native open and save dialogues for Windows users
- Fixed wrong handling of scale variations in Drizzle case
- Fixed 8-bit images auto-stretch display
- Fixed BMP support
- Fixed issues in PNG and TIFF 8-bit export
- Fixed the "About" OS X menu.

1.2. Work on a sequence of converted images

Once you have your Siril's format .fit files, you can switch to the `Sequence` tab seen below.



Sequences

Sequences are what Siril uses to represent the manipulated files.

Three file types are natively supported:

a set of two or more FITS files, a single SER file or, to some extent, a regular film file such as AVI or any other container.

Film files support is being dropped in favour of SER, but simple sum stacking [p.41] can still be done on them.

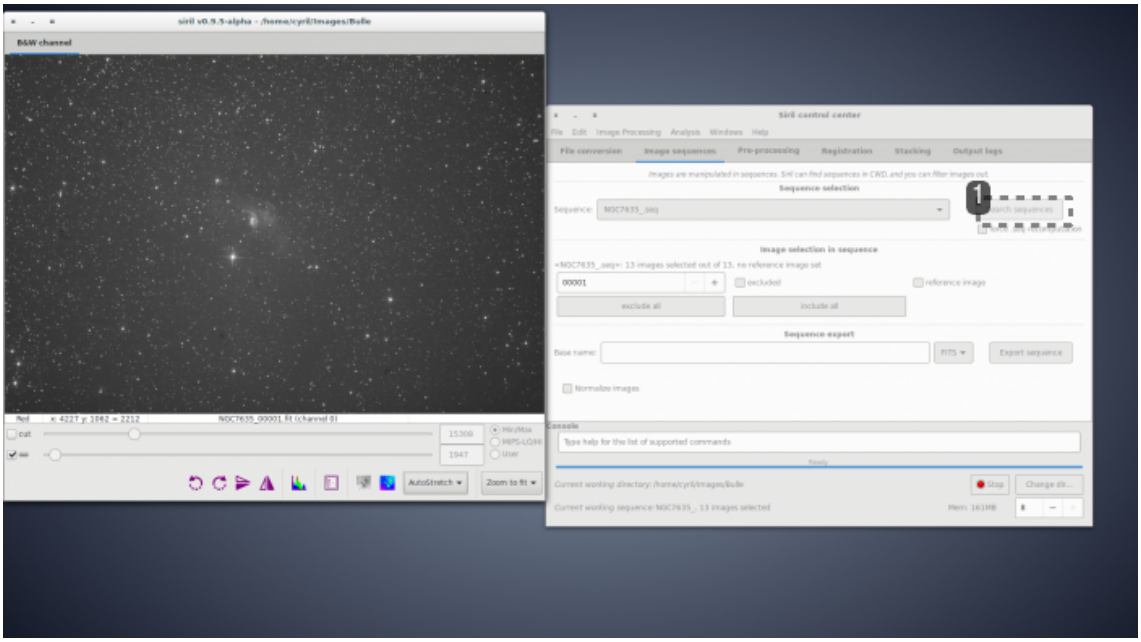
The file extension for FITS files can be configured in the settings, by default it is `.fit`. Changing it will change the extension used to detect FITS sequences in the current working directory, but also the extension of the files created by Siril in conversion or any processing that creates an image.

Click on the `search sequences` button.

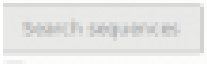
If you only have one sequence in the working directory, it is automatically selected and loaded.

Else, the list will pop-up and you'll be able to select the sequence you want to use.

When the sequence is loaded, two windows may be opened: the grey frame control window, on the left of the image below, and the RGB composition rendering window, behind the main window on the image below. In the case of monochrome images or CFA files, only the grey frame window will be opened.



1



search sequences

The grey window is used to display color components for the image, control their visual composition in the color image, and give information about the displayed frame, pixel value when moving the mouse on a pixel, file name and layer number, zoom value, FWHM[p.189] value of the selection if it applies, and some display tools.

1.3. Registration

Registration

Registration is basically aligning the images from a sequence to be able to process them afterwards.

Calculate shift in images to be aligned with the reference image

Images are not modified, only shift parameters are saved in `regparam` in the sequence.

Layer is the layer on which the registration will be done, green by default (set in

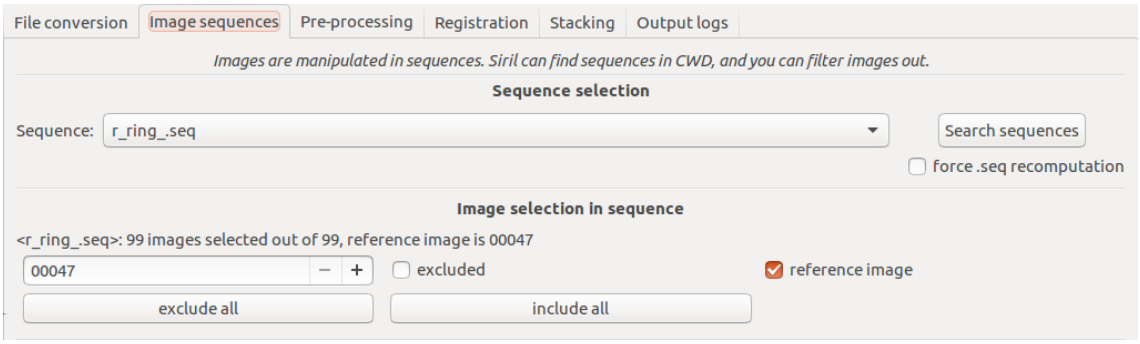
> *siril_init()*)

If in Images Sequence tab reference image is set :

The current image will be the reference image for registration.

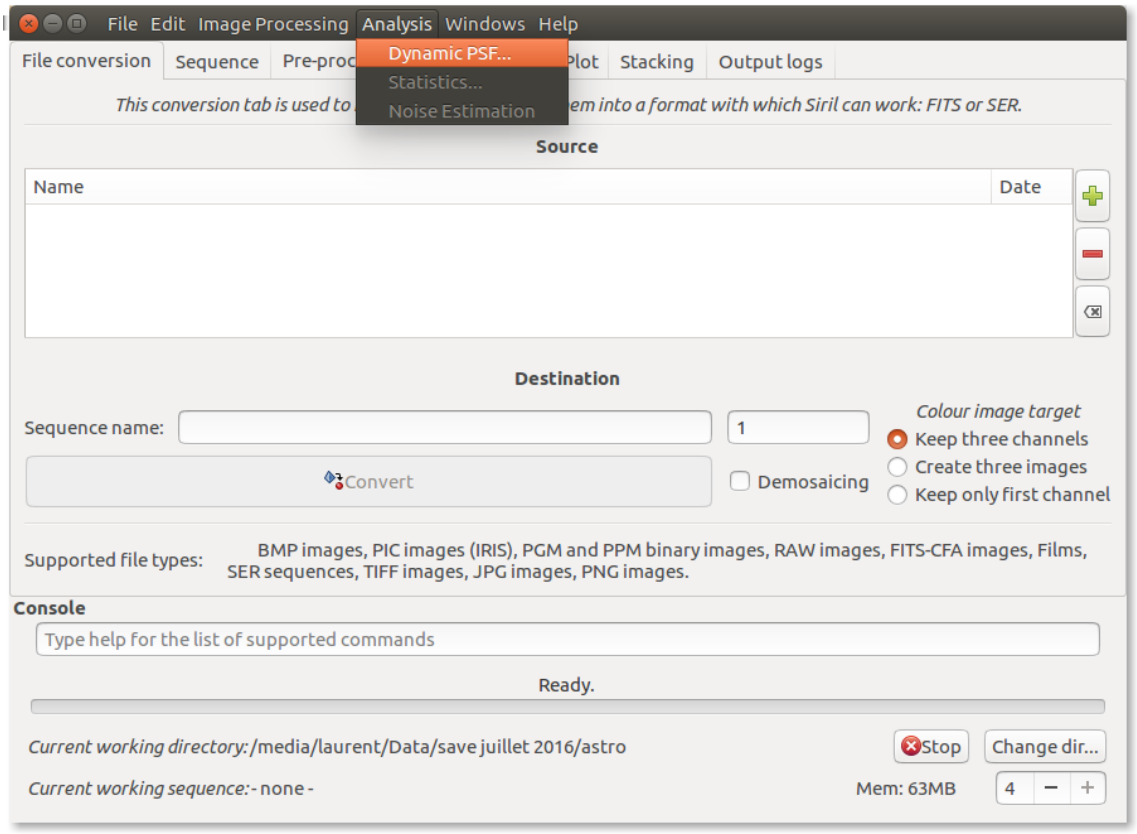
Only one image can be the reference image in a sequence, setting it will unset the image previously set as reference.

If not set manually, the first image of the sequence is taken as reference.



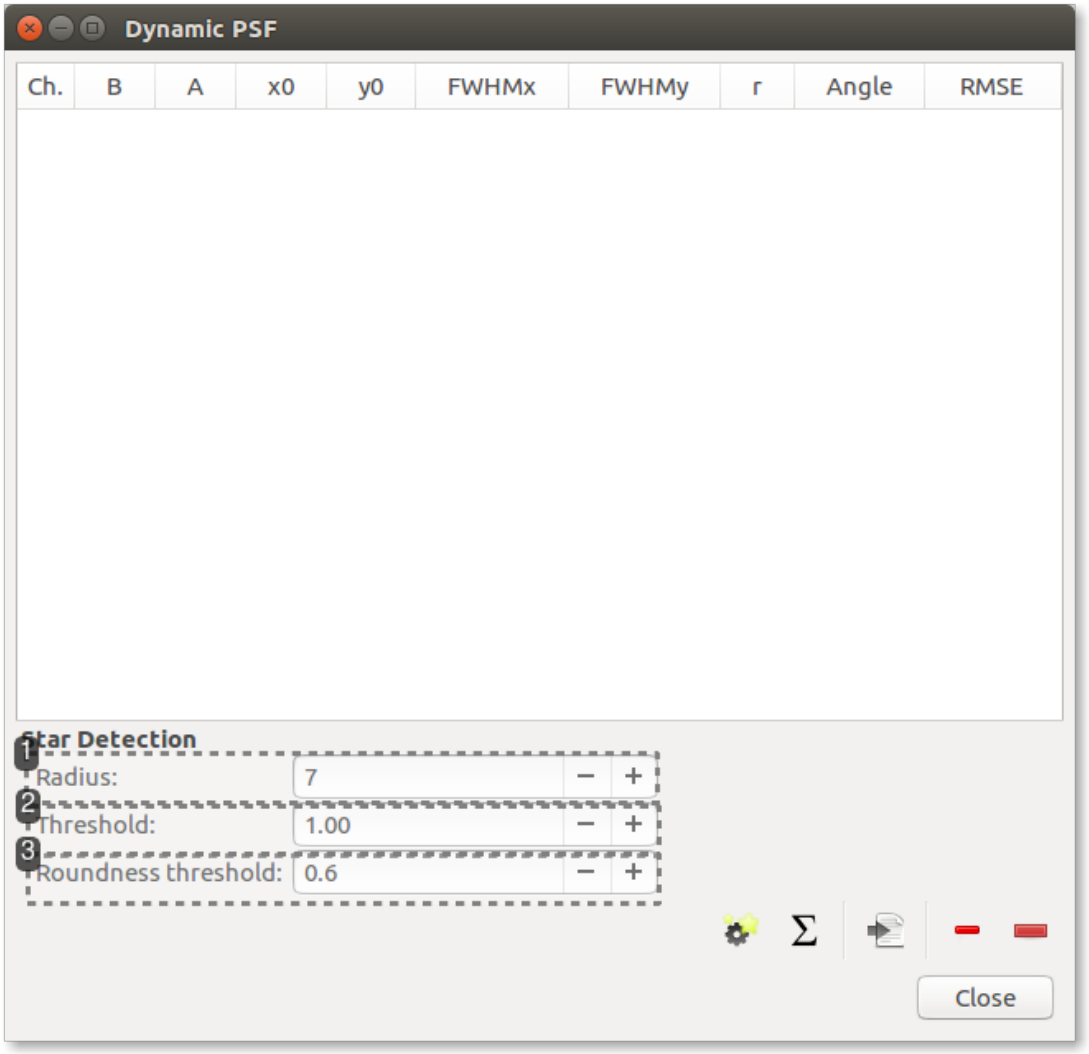
Adjust Registration settings

Open ☰Analysis/Dynamic PSF...



Dynamic PSF (for registration settings)

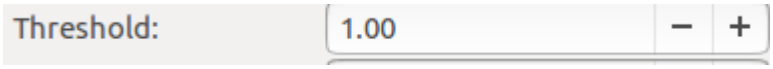
Adjust registration settings



1 Radius

Size in pixels of the search box.

2 Threshold value

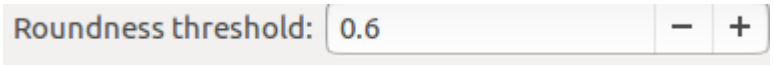


Threshold value above the noise for stars detection.

The median and standard deviation values of the image are determined, then the threshold is: `median + [threshold] x stdev`.

> The default value for threshold is 1.00.

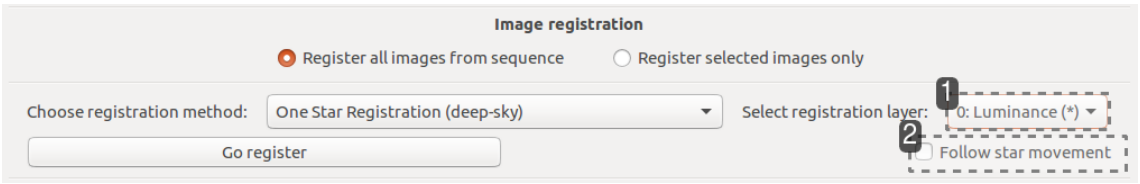
3 Threshold value



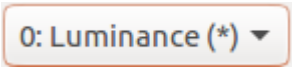
Threshold value of star roundnesses. Decrease this value if your stars show small trails.

> Ensure you have a good reason to change this value because it may degrade the star detection.

1.3.1. One Star Registration (deep-sky)



1



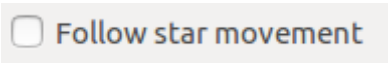
Layer on which the registration is done.

It can only be done on one layer at a time, so on monochrome images.

> The (*) sign appearing after the layer's name means that registration data is already available for this layer.

> When processing images, registration data is taken from the default layer if available (for RGB images: Green, else fallback to Blue then Red).

2 **Follow star movement**



Use this option to use the position of the star found in the previous image as new center for the current image registration.

This allows the selection area to be smaller, registration faster, and accounts for drift or images with a large number of stars.

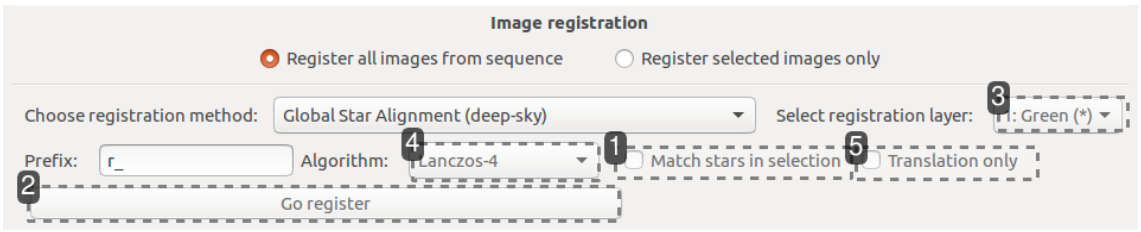
> Enabling this option requires the registration to not be parallelized, it will run on one CPU core only.

This is the simplest method to register deep-sky images.

Because only one star is concerned for register, images are aligned using shifting (at a fraction of pixel). No rotation or scaling are performed.

Shifts at pixel precision are saved in seq file.

1.3.2. Global Star Alignment (Deep-sky)

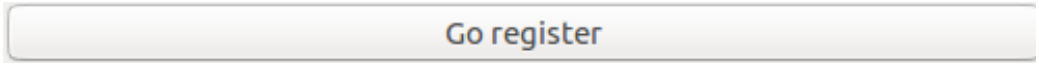


1

Match stars in selection Check this box if you want to perform the Global Star Alignment algorithm within the selected area in the reference image. If no selection are done, this option is ignored.

Make sure to draw a selection with a sufficient number of stars.

2



Start the registration process

It will register images in the loaded sequence using the selected method and reference layer.

Loaded in the 'Image sequences' tab, current sequence name is at the bottom of the window too.

3

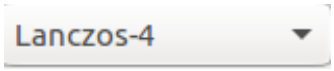
Layer on which the registration is done.

It can only be done on one layer at a time, so on monochrome images.

The (*) sign appearing after the layer's name means that registration data is already available for this layer.

When processing images, registration data is taken from the default layer if available (for RGB images: Green, else fallback to Blue then Red).

4



Type of algorithm used for rotation

Choose algorithm:

- Nearest Neighbor
- Bi-linear
- Pixel Area Relation
- Bicubic
- 👉Lanczcos-4

5 **Translation only**

This is the only registration that can detect and apply rotations on images. However, rotating images will degrade them to a certain extent.

Activating this will use the same image alignment algorithm but only apply one-pixel accurate translations (rotation is computed, but not applied).

This is useful for image sequences with very small rotation angle.

no new sequence is built, registration data is stored in the original sequence.

Since version 0.9.7, Siril uses a new and more robust alignment algorithm than previous releases.

Any automatic image alignment algorithm (either for panorama or Astrophotography purposes) must detect relevant image features, known as control points.

These points must be matched by correspondance on two images (reference and target) in order to determine the transformation that a software could apply. Luckily, in deep sky astrophotography, stars are nearly ideal control points.

Siril alignment works as follow:

- Stars are detected in the reference image. The number is displayed in the Siril consol.
- Stars are detected in the target picture, similarly their number is displayed in Siril consol.
- An algorithm is building triangles from the brightest detected stars in the reference image, then there are associated to the corresponding triangles in the target one.
- A first order transformation is computed and stars are matched between them in the two images. The number of matched stars is shown in the Siril consol.
- The two lists of stars are processed with a new algorithm named RANSAC[https://en.wikipedia.org/wiki/Random_sample_consensus]. This algorithm is abled to deal with a large fraction of outliers in order to only keep good stars.

This is a 3 x 3 matric with 8 degrees of freedom like:

$$H = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & 1 \end{bmatrix}$$

This algorithm makes it possible to ignore the stars which it considers unreliable in order to determine a new more precise transformation, expressed in the form of a homographic matrix[https://en.wikipedia.org/wiki/Homography].

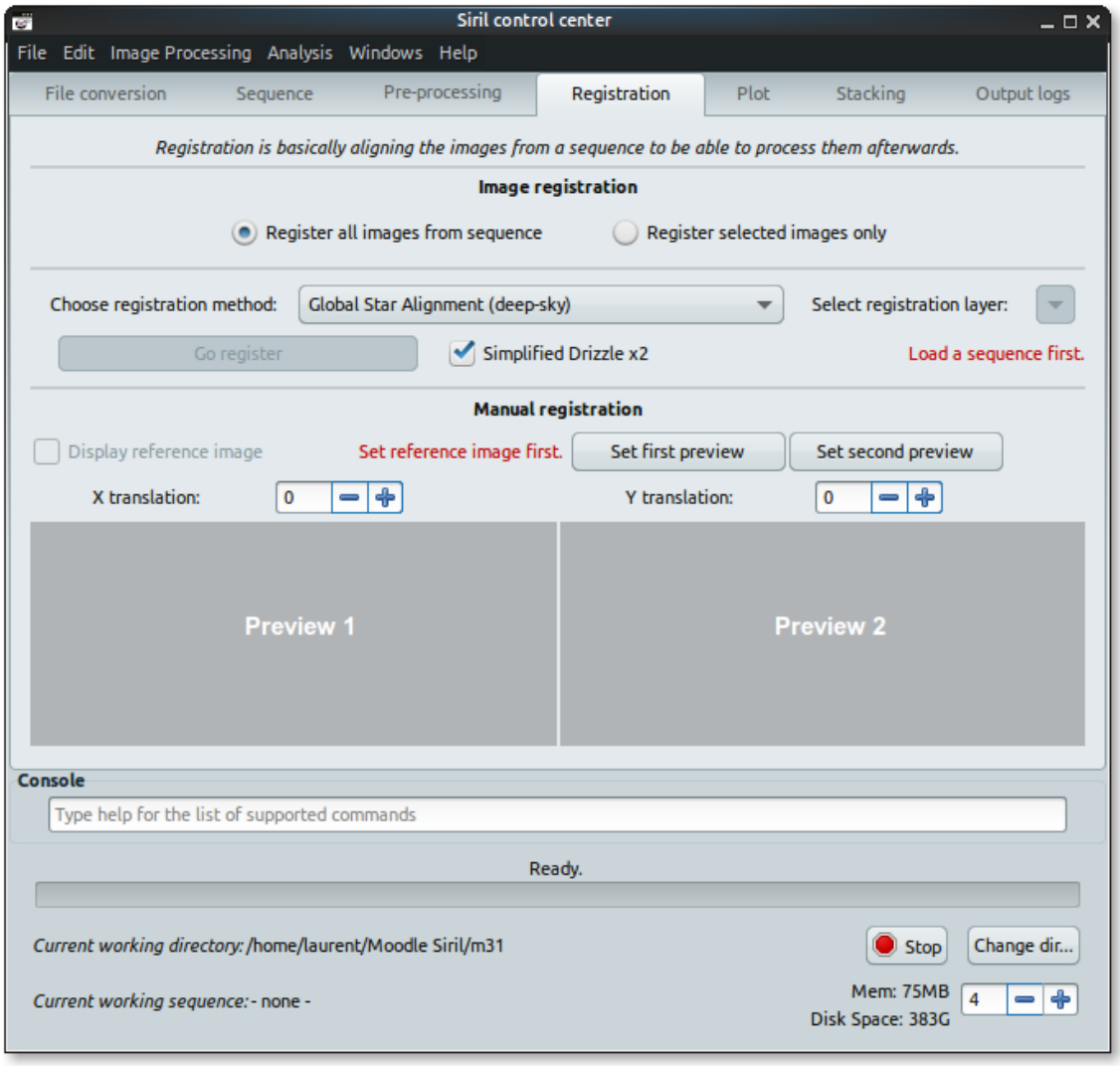
When G = H = 0, it is an affine transformation, or linear, which is often the case.

GIMP 2.9 has a good use of the homographic matrix in its "Unified Transform Tool".
You can easily how the transformation will change matrix parameters.

A new sequence is created with the prefix of your choice (r_ by default).

Since Siril 0.9.7, Images registration can be applied to images with different focal lengths and/or sizes.

1.3.3. Simplified Drizzle x2



Simplified Drizzle x2

Activate the simplified drizzle algorithm for the processing of this sequence.

An up-scale (x2) will be applied to the registered frame or during stacking depending on which registration is chosen, that will result in higher resolution images.

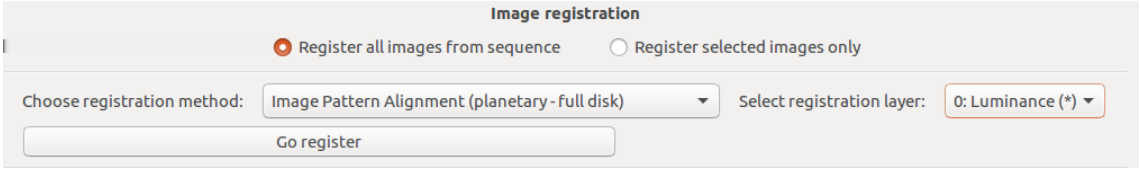
This option is adapted for under-sampled images, i.e, when the telescope focal length is too short for the pixel size.

One may consider that the system is under-sampled when FWHM is smaller than 2 pixels.

The correct name of this method should be super-resolution stacking, but for a more convenient understanding we called it ***Simplified Drizzle x2***

> the counterpart of this technique is that the amount of memory and disk space needed to create and process drizzled images is multiplied by the square of the Drizzle factor.

1.3.4. Image Pattern Alignment (planetary-full disk)

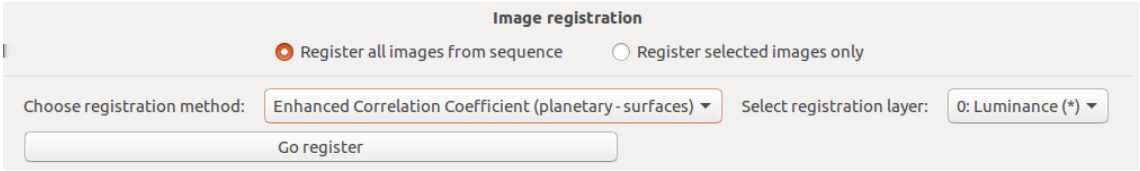


This is a simple registration by translation method using cross correlation in the spatial domain.

This method is fast and is used to register "planetary movies. It can also be used for some deep-sky images registration.

Shifts at pixel precision are saved in seq file.

1.3.5. Enhanced Correlation Coefficient Maximization (planetary-surfaces)

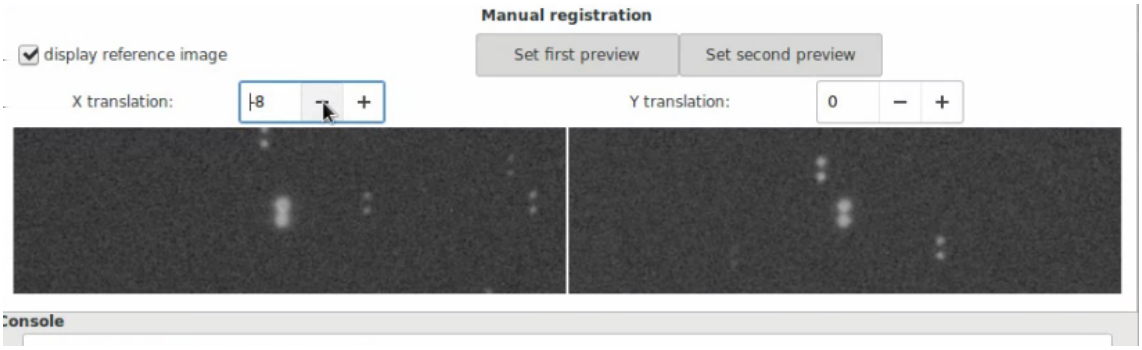


It is based on the enhanced correlation coefficient maximization algorithm.

This method is more complex and slower than Image Pattern Alignment but no selection is required. It is good for moon surface images registration.

Although the rotation is possible with this algorithm, but it was currently decided to only allow a detection of translation.

1.3.6. Manual Registration



In some case manual registration is very useful for a few frames

The reference image is displayed with a transparency of 50%, it must be selected before manual alignment.

Then select an image area with a single click in it for translation in x and another for the y

> The mouse wheel or up or down and left or right arrows can be used to perform the translation in x or y. Simply select first the axis of movement by clicking in X translation or Y translation entry field when the mouse wheel is used.

1.4. Plot

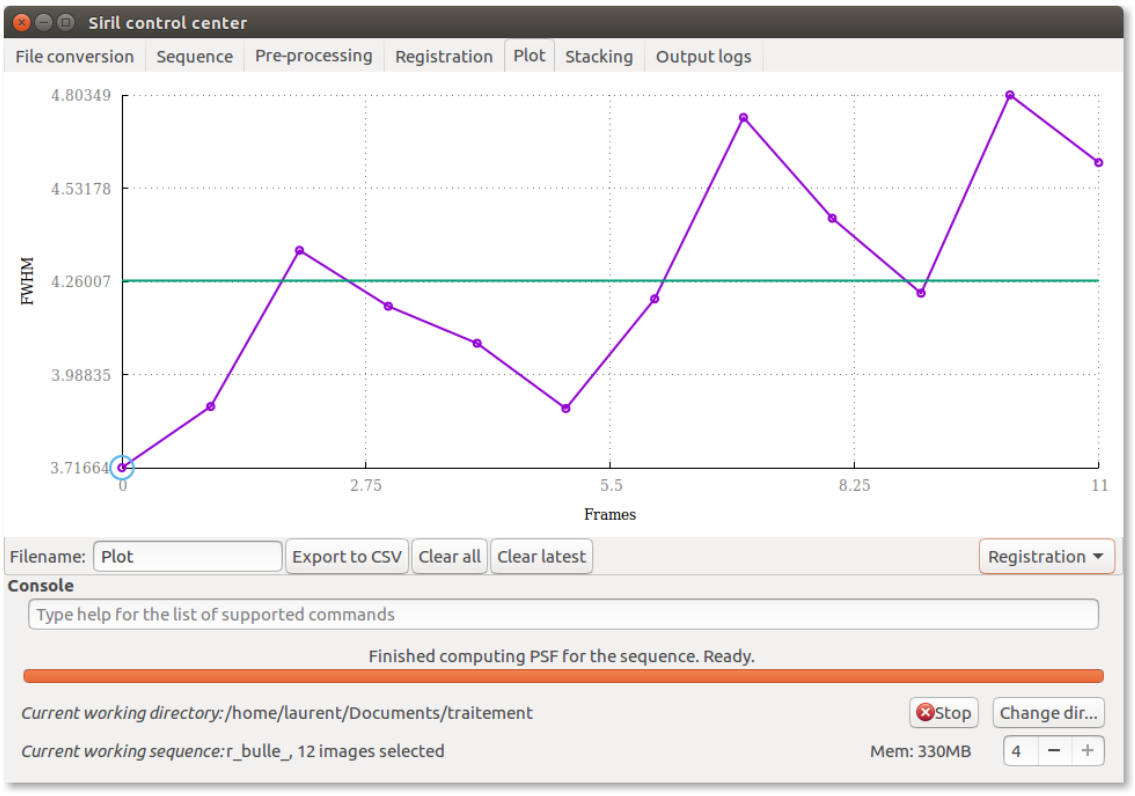
A new, very useful function has just been added

- Main use : you can select with many criteria the images to be stacked.
- Or use the data by exporting csv for many other purposes like photometry

> **Use of csv data export**
>
> It is very useful for observing variable stars, like Eclipsing binary of Algol type.

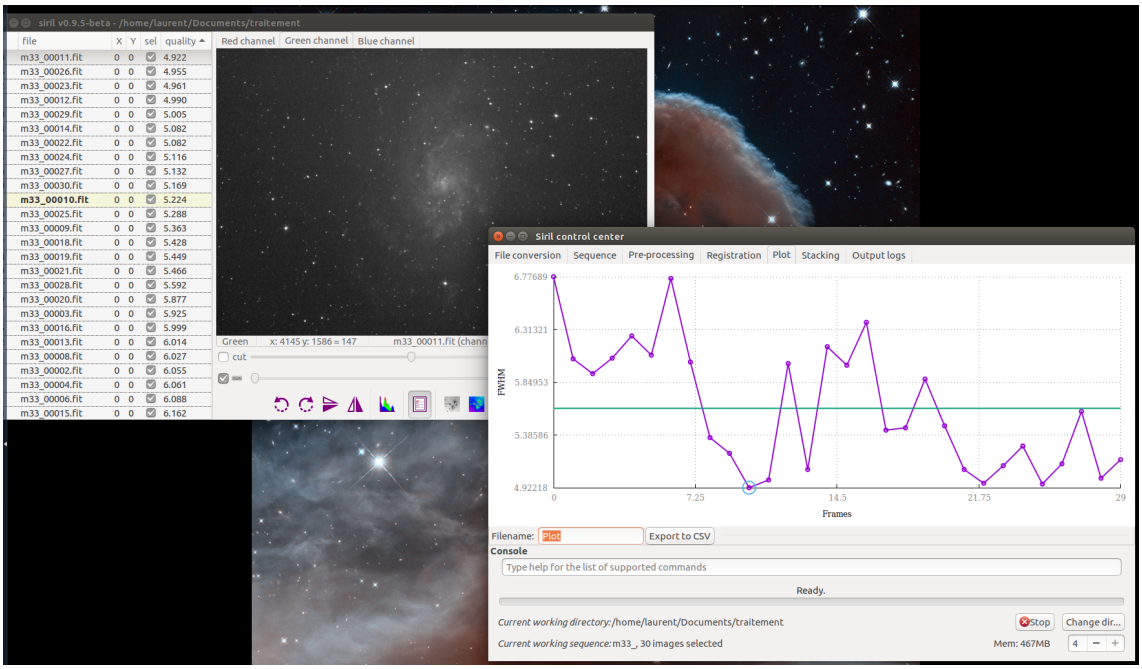
Default view

After a first registration a graph is displayed



The reference image is surrounded

You can select a reference image before computing a new alignment with reference to it



Use show/hide list of images in the sequence with registration data in gray channels images windows to select or deselect images

The current image is selected by a double click in the list

The reference image is selected in the Sequence tab

Get more information

You can also choose to have more information to set your reference image or remove images

Get more information from a set of images

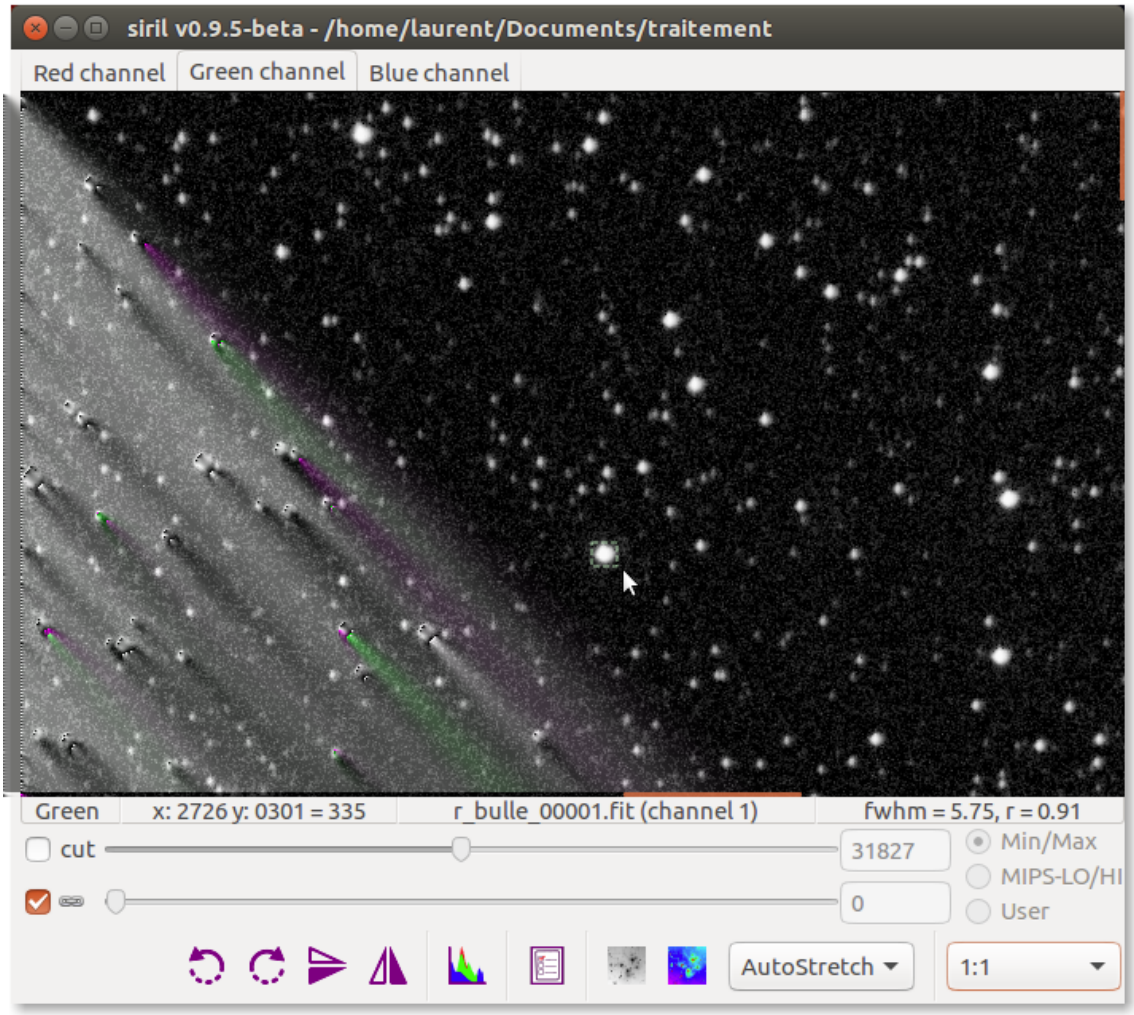
When you use this feature for other purposes than stacking, it is more meaningful to obtain additional information about a sequence, if it has been normalized.

No registration is necessary, simply load the sequence

1. Pick up a star

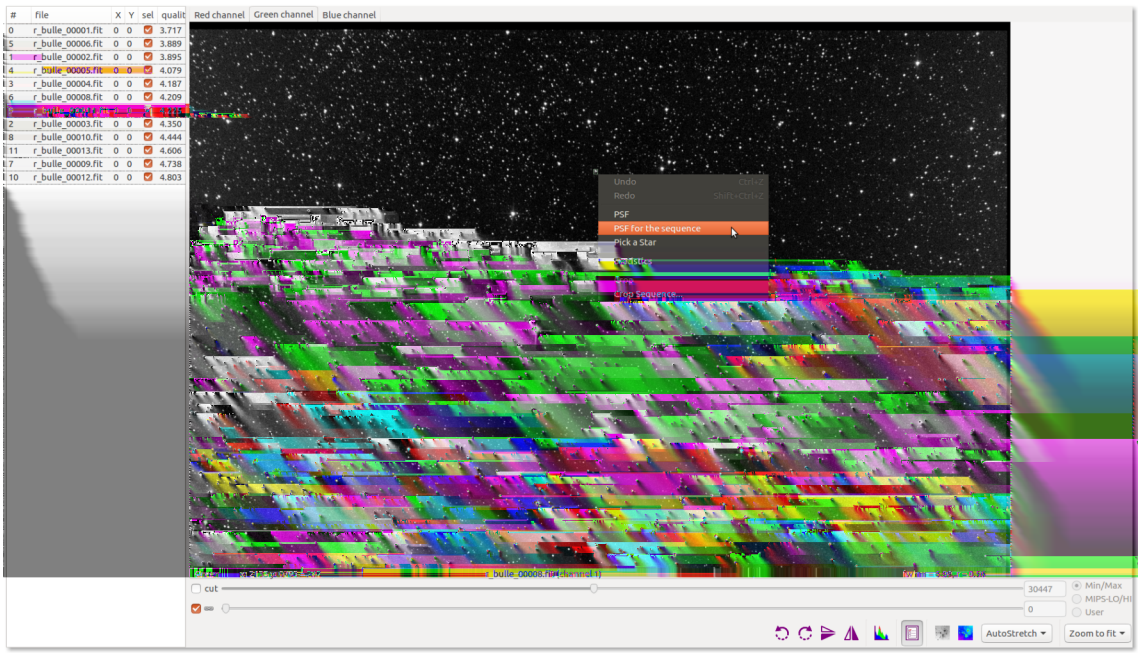
   In Gray Channels images windows

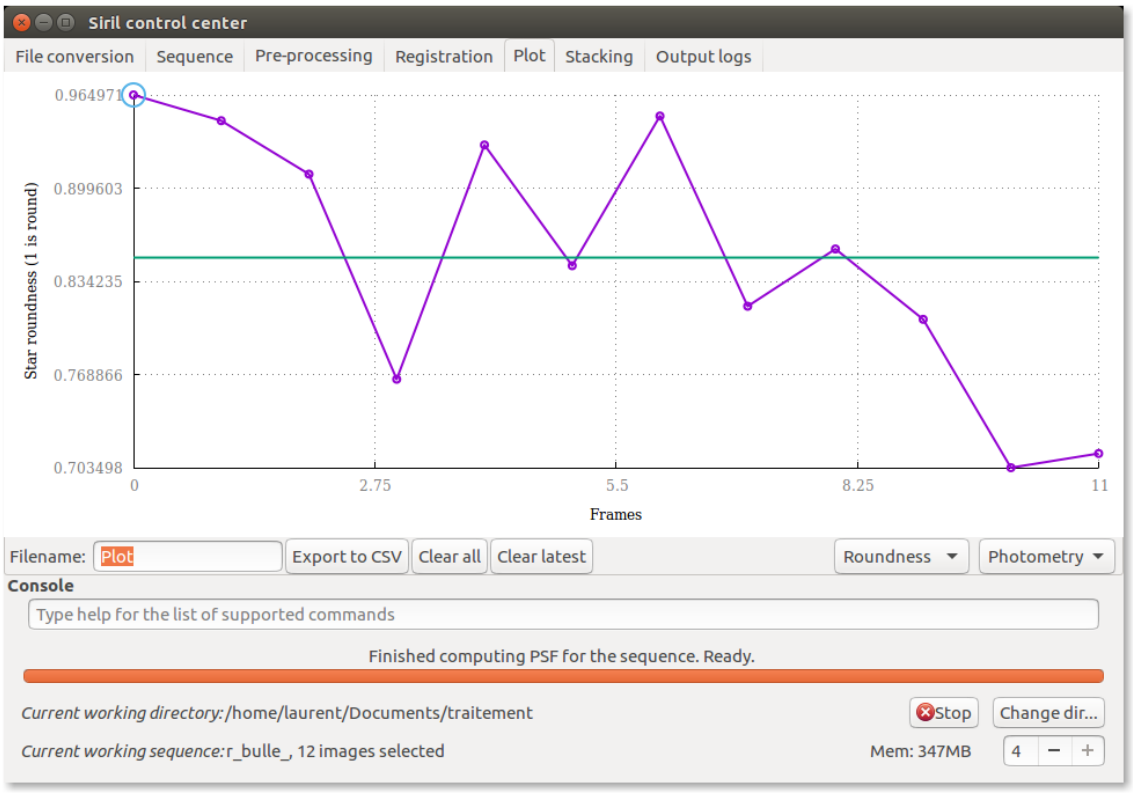
   Draw a box around the star



   Use zoom level for more accuracy

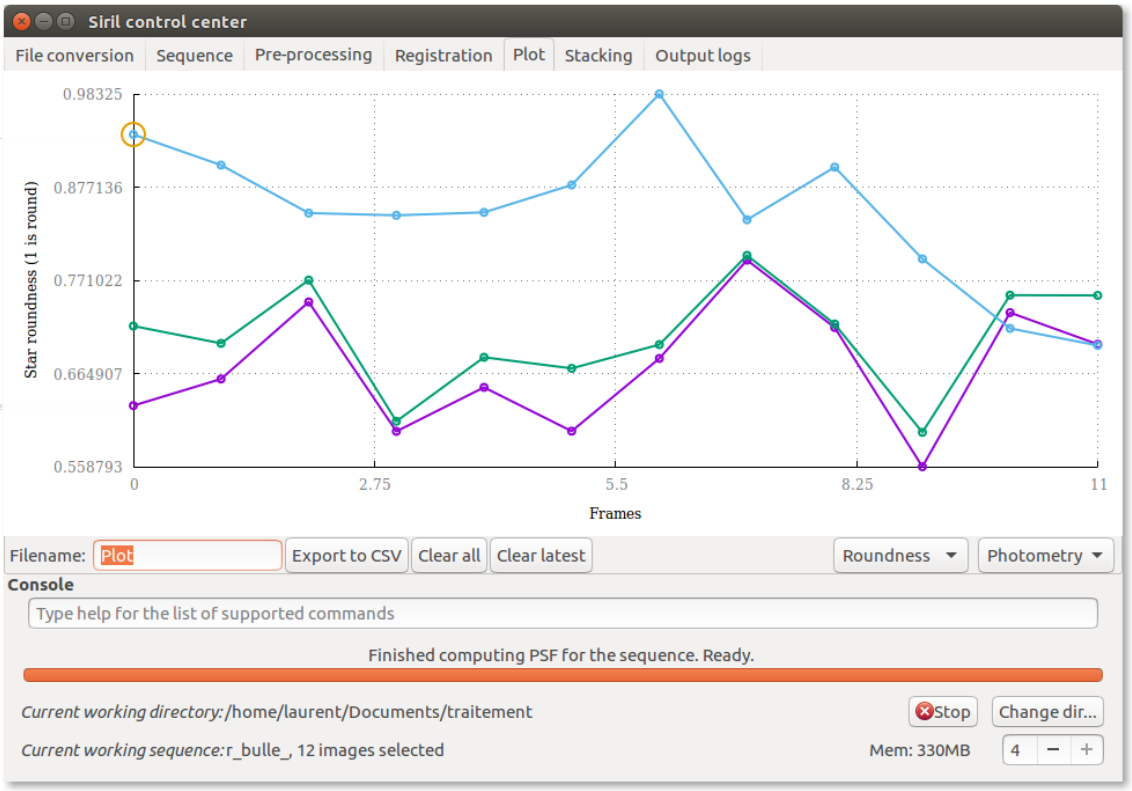
2. Right click and choose *PSF for sequence*





3. Repeat the previous step until you have a representative sample of stars

Result

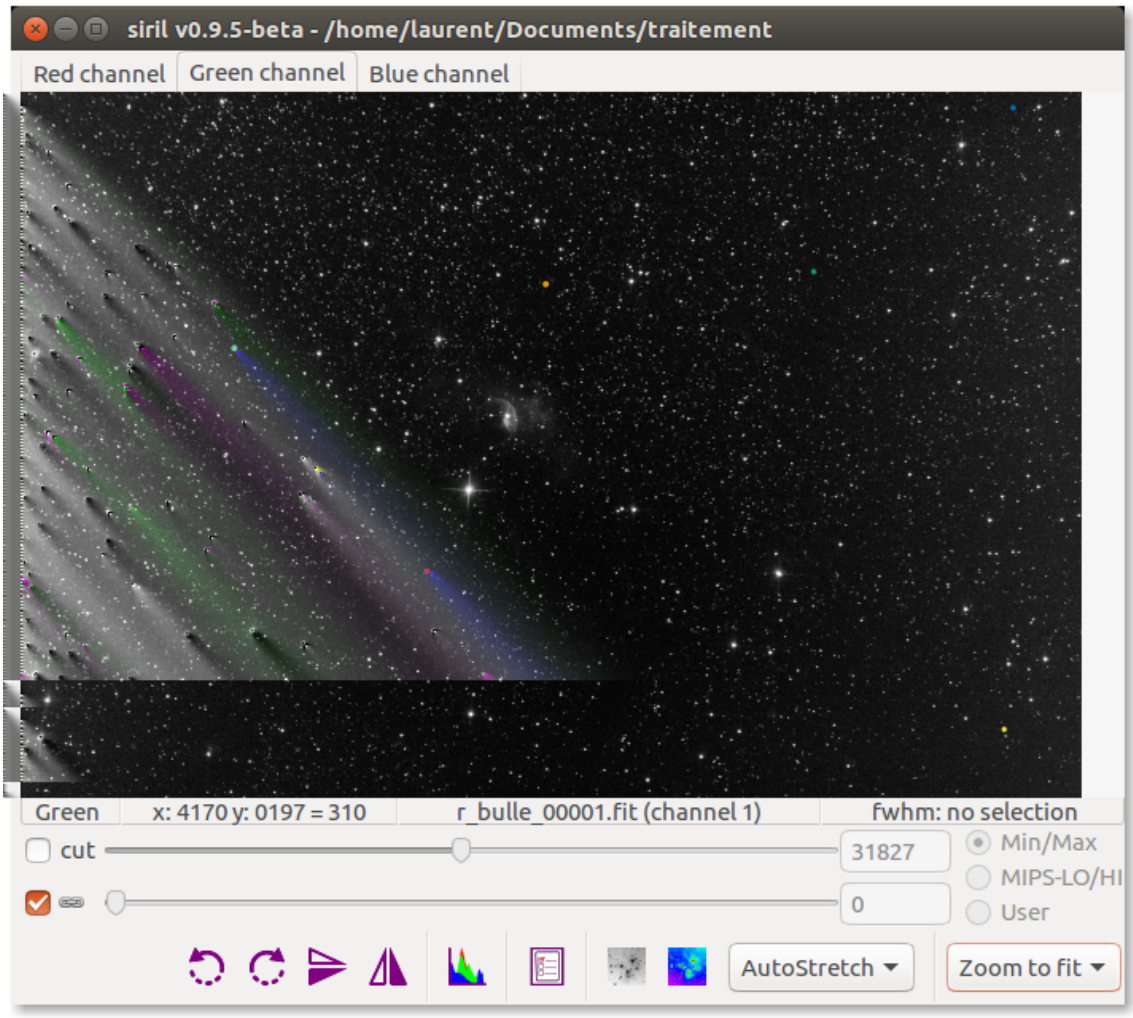


General comments

Now you have the following information to optimize your choices for the selected image or the reference image:

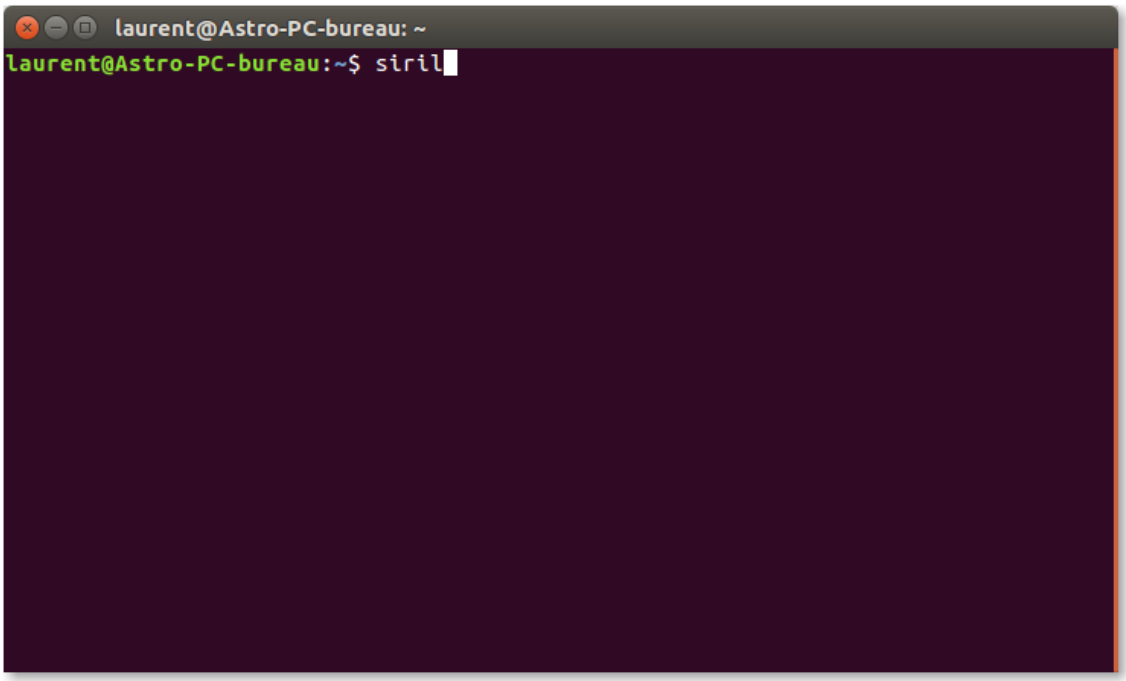
- Roundness
- FWHM
- Amplitude
- Magnitude
- Background
- Star position on X axis
- Star Position on Y axis

Each star is identified as a circle which color is identical to the color of the corresponding curve in the graph.



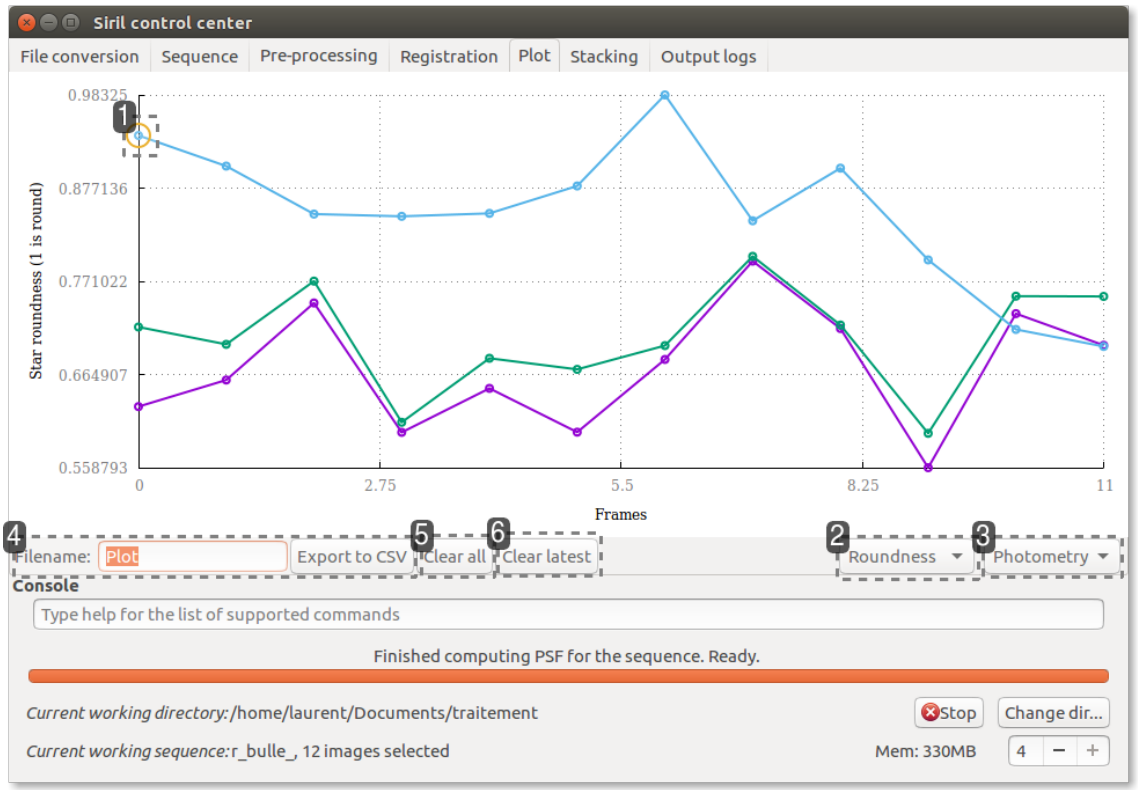
Additional data

When Siril is launched in command line from a Linux console



Additional data is displayed in the console as the position of the selection and its size in pixels

```
laurent@Astro-PC-bureau: ~/siril
log: See the console for a dump of star data over the sequence (stdout)
# image_no amplitude magnitude fwhm x y
Purging previously saved reference frame data.
Purging previously saved reference frame data.
Purging previously saved reference frame data.
selection: 4127,1842,   1x0
selection: 4236,1766,   19x18
log: Running the PSF on the loaded sequence, layer 1
log: Results will be displayed at the end of the processing, on the console outp
ut, in the following form:
log: Starting sequence processing of PSF
log: Loaded partial FITS file r_bulle_00001.fit
log: Loaded partial FITS file r_bulle_00002.fit
log: Loaded partial FITS file r_bulle_00003.fit
log: Loaded partial FITS file r_bulle_00004.fit
log: Loaded partial FITS file r_bulle_00005.fit
log: Loaded partial FITS file r_bulle_00006.fit
log: Loaded partial FITS file r_bulle_00008.fit
log: Loaded partial FITS file r_bulle_00009.fit
log: Loaded partial FITS file r_bulle_00010.fit
log: Loaded partial FITS file r_bulle_00011.fit
log: Loaded partial FITS file r_bulle_00012.fit
log: Loaded partial FITS file r_bulle_00013.fit
log: Finished sequence processing of PSF
```



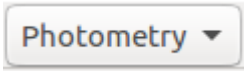
1 Selected reference image

The reference image is surrounded

2 Select which information

- Roundness
- FWHM
- Amplitude
- Magnitude
- Background
- X position
- Y position

3 Information



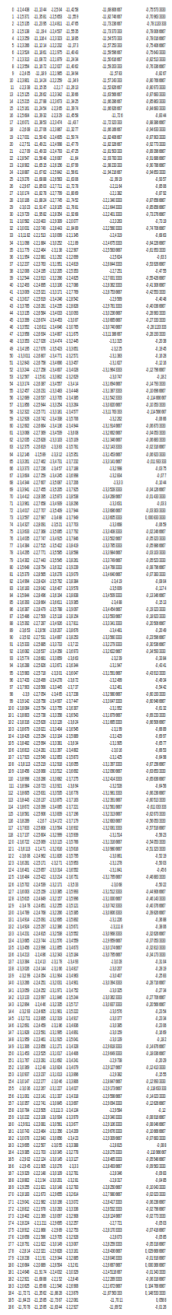
Select the source of graph information.

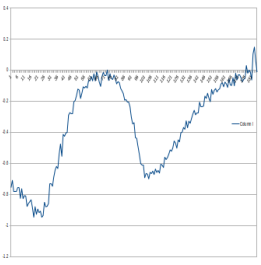
- Photometry
- Alignment

4 Export

Export graph to csv

Exploitation of a csv export: Cy Aqr2



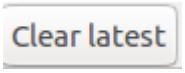


5 Clear all

Clear all

Reset graph data

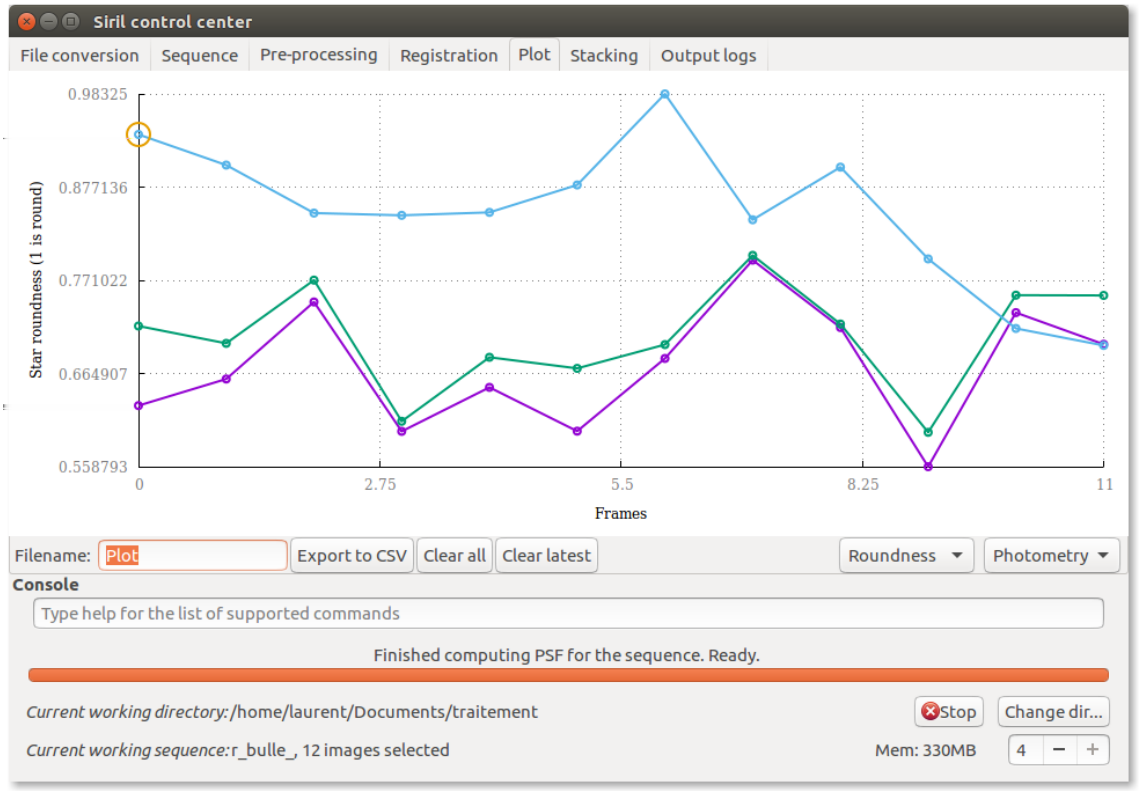
6 Clear latest



Erases the last curve

The model computes a Gaussian task with two axes, the major axis X and the small axis Y for the slightly elongated stars.

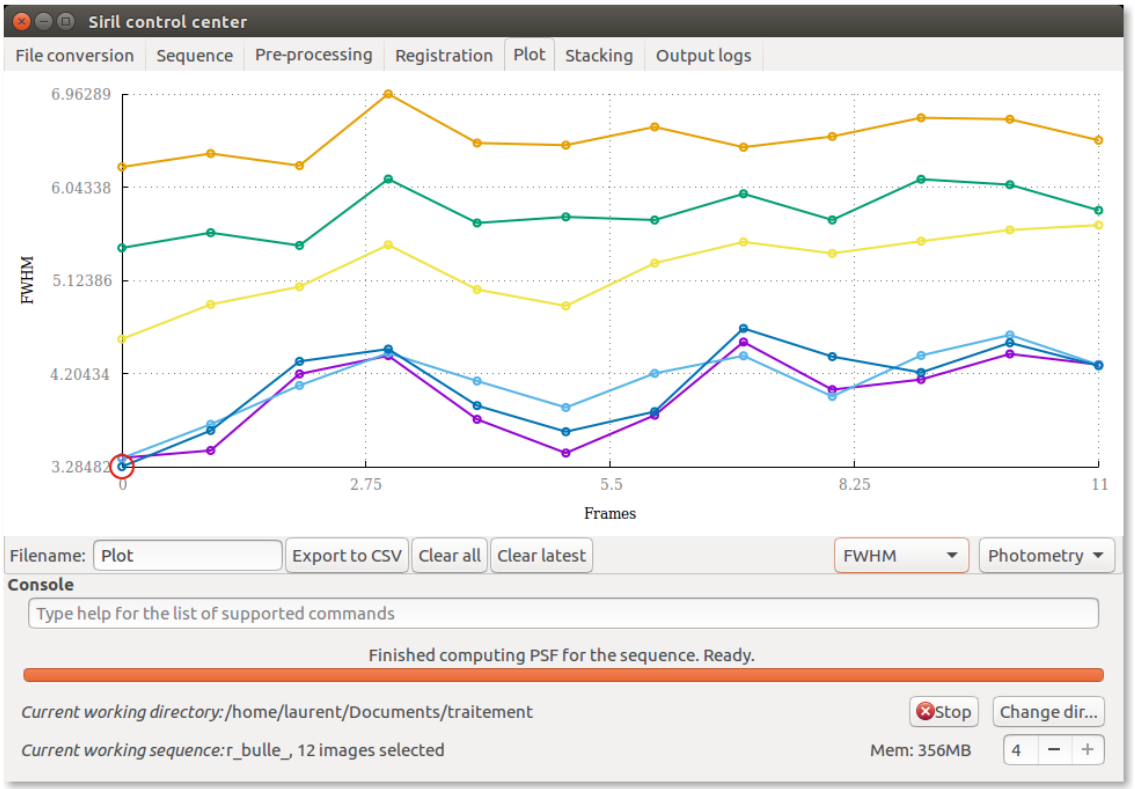
Roundness



The roundness parameter. It is expressed as fwhm_Y /fhwm_X

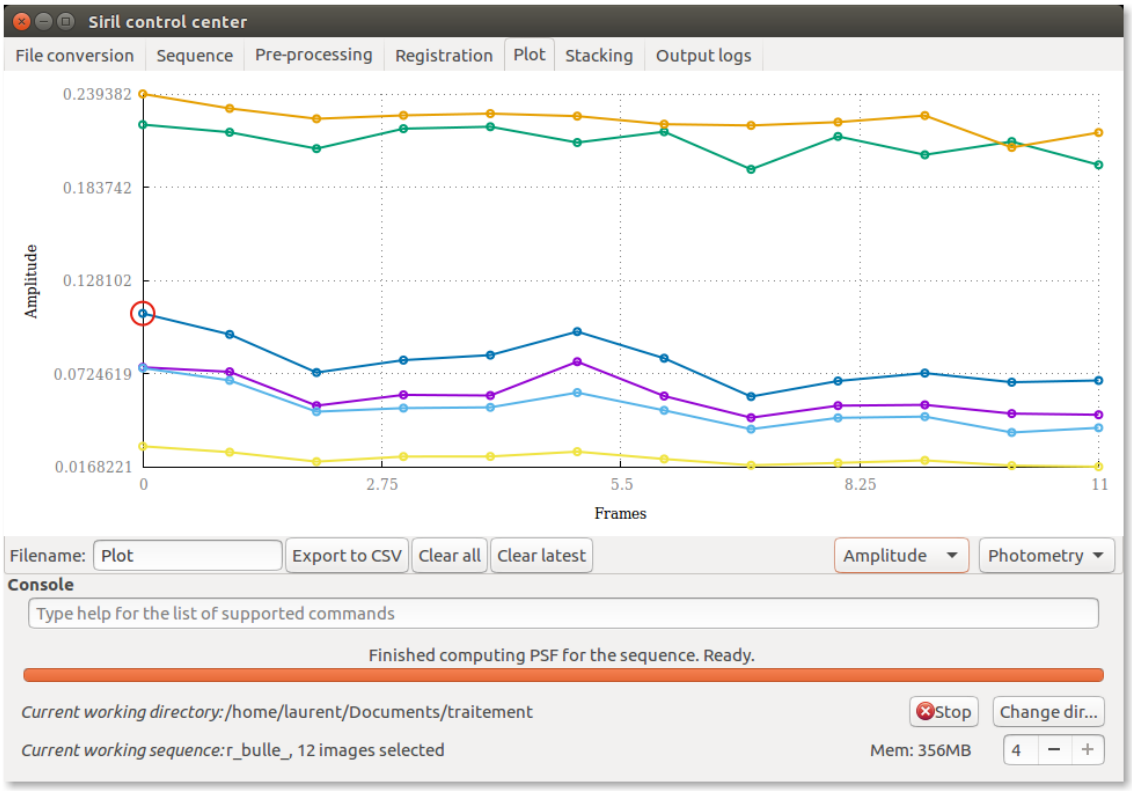
The range is 0 to 1

FWHM



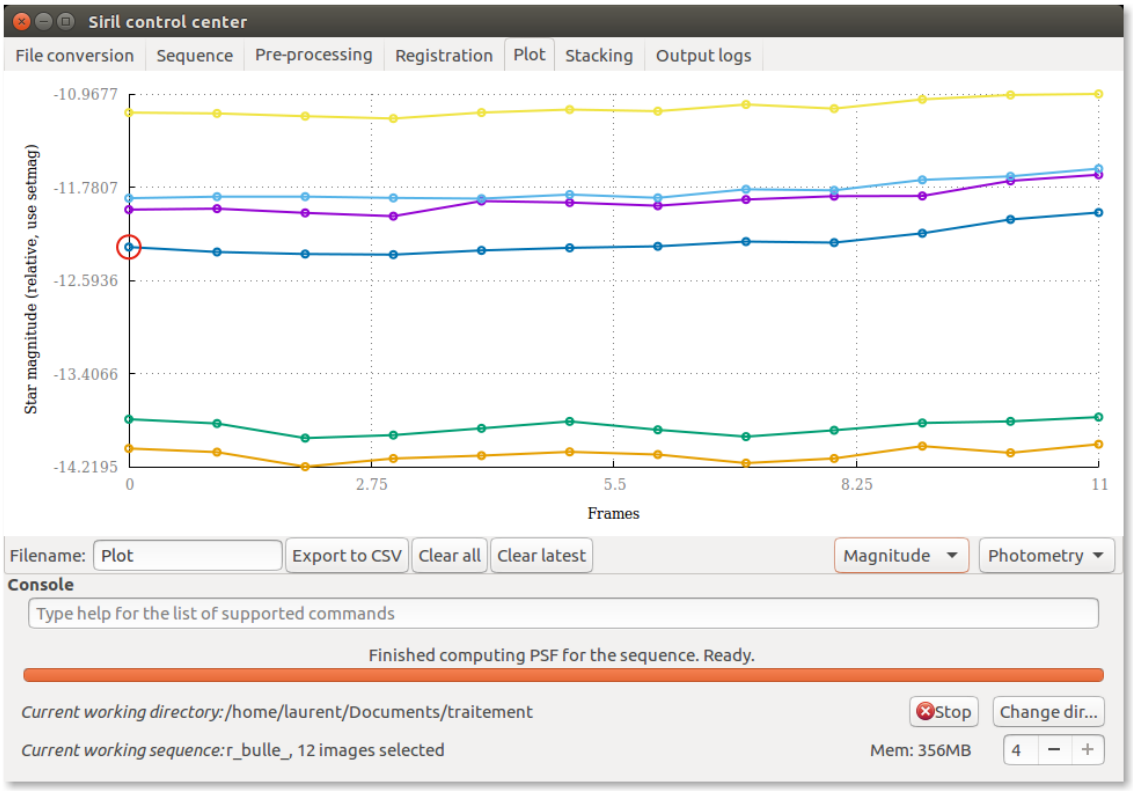
The Full Width Half Maximum[p.189]

Amplitude



The maximal intensity of the star in the [0, 1] range: it is the A parameter in 2d Gaussian[p.183] model. It is the peak value of the fitted function, located at the centroid coordinates x0 and y0.

Magnitude



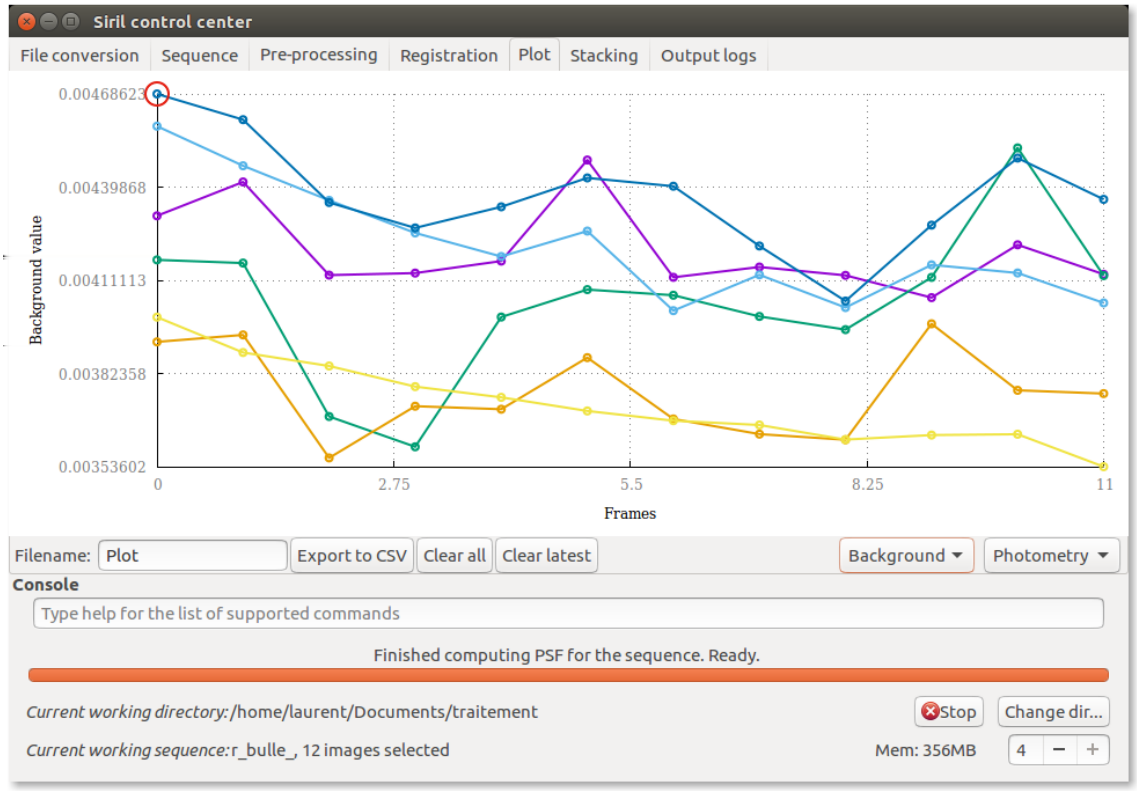
The magnitude is the integral of the Gaussian

Without Calibration with `setmag` It is given in ADU units[p.183]

You can 👉normalize your sequence and use the command 👉setmag to calibrate the selected star's relative magnitude

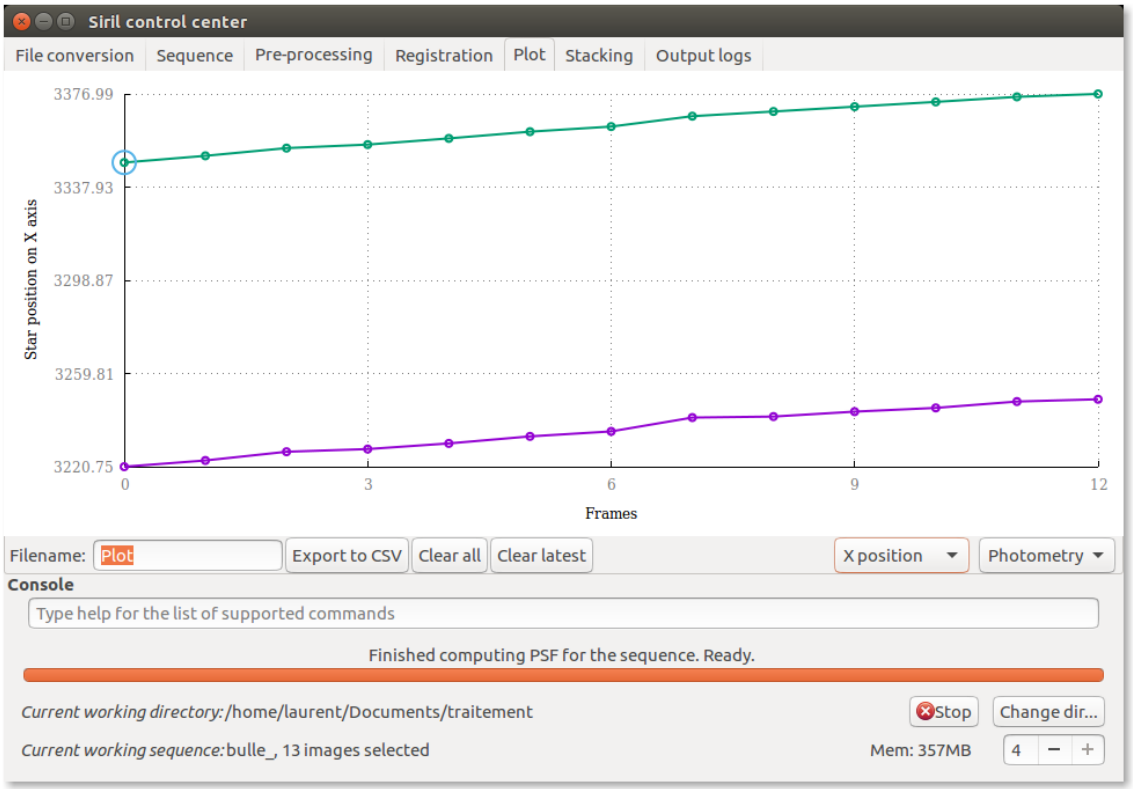
`setmag` associates an ADU integral with an apparent magnitude value and allows to express the apparent magnitude of the other images. It is just an offset.

Background



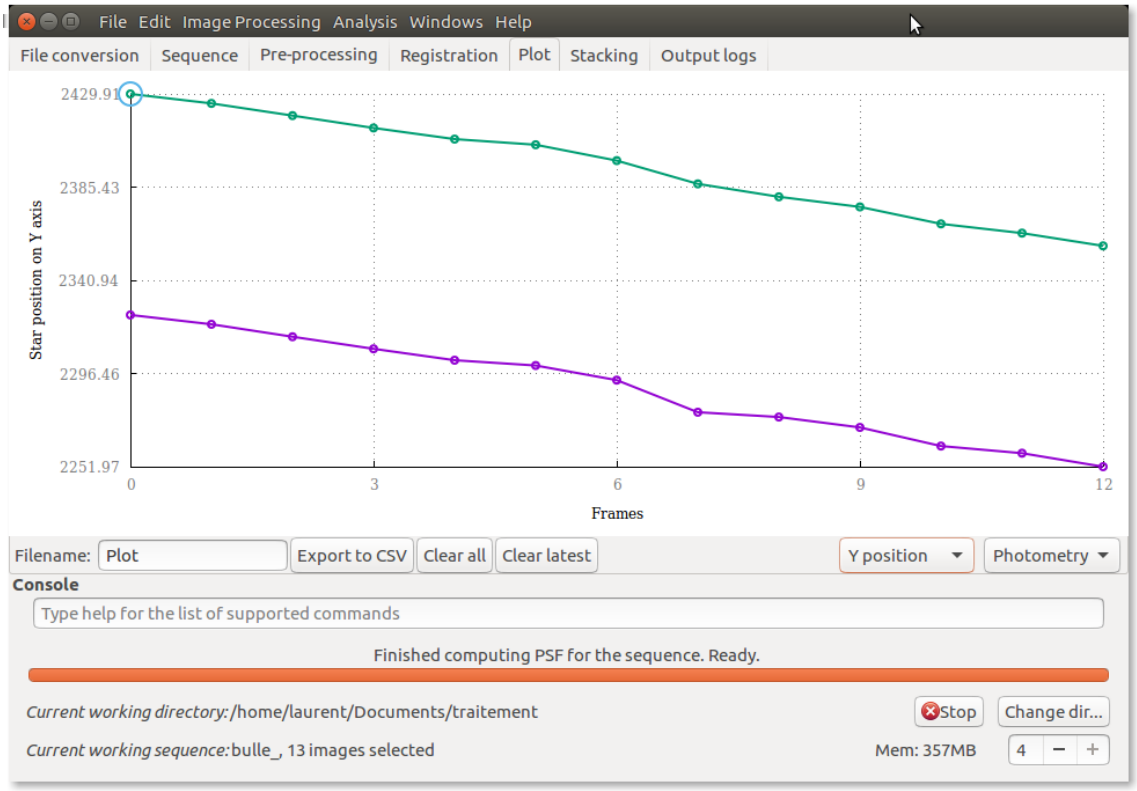
The sky background is an average of the sky background pixels around the star or the level from which the gaussian starts.

Star position on X axis



Do not make registration

Star position on Y axis



Do not make registration

You can use these data to calculate the mount's periodic error curve with the appropriate sequence.

csv data export

You can export data of an images sequence

Order of columns: number of the image, sample n to n+x

0	0.00391868	0.00409481	0.00419305	0.00402577	0.00366996
1	0.00386159	0.00411887	0.00415182	0.00407796	0.00367079
2	0.00383173	0.00407711	0.00406276	0.00400722	0.00358929
3	0.00375529	0.00401785	0.00405879	0.00399004	0.00354695
4	0.00372182	0.00397001	0.00399752	0.00395035	0.00343857
5	0.00368188	0.00391017	0.00393777	0.00389273	0.00348703
6	0.00362026	0.00386605	0.00388906	0.00385509	0.00355442
7	0.00357674	0.00381525	0.00379023	0.00378899	0.00339784
8	0.00357034	0.00378527	0.00378296	0.0037729	0.00338058
9	0.00356047	0.00378576	0.00384871	0.00382624	0.00352673
10	0.0035689	0.00378053	0.00379046	0.00379193	0.00353248
11	0.00351931	0.00371972	0.00372365	0.00373539	0.00349799

You can export data for :

- Roundness
- FWHM
- Amplitude
- Magnitude
- Background
- X Star position
- Y Star position

The csv file is saved by default in the current working directory

Use Stellarium[http://www.stellarium.org/] or Kstars[https://edu.kde.org/kstars/] in association with astrometry.net[http://astrometry.net/] remotely[http://nova.astrometry.net/] or locally to identify the stars

Setmagseq

setmagseq

`setmagseq magnitude`
This command is only valid after having run seqpsf or its graphical counterpart (select the area around a star and launch the *`psf analysis`* for the sequence, it will appear in the graphs).

This command has the same goal as *`setmag`* but recomputes the reference magnitude for each image of the sequence where the reference star has been found.

When running the command, the last star that has been analyzed will be considered as the reference star.
Displaying the magnitude plot before typing the command makes it easy to understand.

To reset the reference star and magnitude offset, see *`unsetmagseq`*.

Example of using the *`Setmagseq`* command

Make a usual *`seqpsf`* to have a curve.

- Display the graph of magnitudes to have a visual feedback
- Type `setmagseq 10` if it does 10 of magnitude
- Take a look to the result

The last star that was studied with *seqpsf* is defined as reference.

Look at the info messages on the console output tab

Undo with *unsetmagseq*

1.5. Aperture photometry

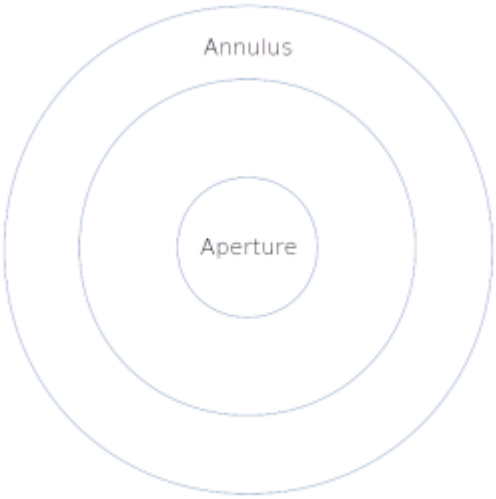
Aperture photometry

Aperture photometry is a technique concerned with measuring the flux of an astronomical object.

In our case, the tool will extract the raw star magnitude that could be compared to the real magnitude as stored in catalogs.

The technique consists of summing all unsaturated and non-zero pixels within an aperture centered on the object.

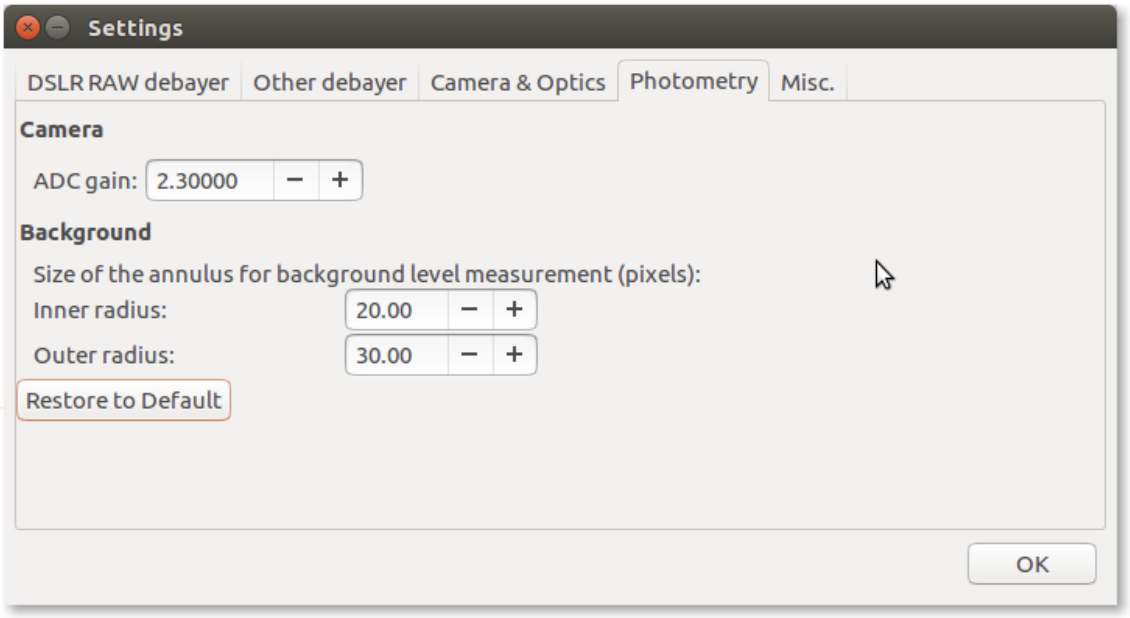
The aperture radius is defined from the **fwhm**[p.189] of the object.
Then the product of the nearby average sky count per pixel and the number of pixels within the aperture are subtracted.



To estimated local background, we use another aperture that has an annular shape.
In ideal case this sky annulus should only contain sky pixels, but a robust mean-algorithm, the same used in the C-Munipack software[http://c-munipack.sourceforge.net/] , rejects outliers.
The annulus is defined with the radii of two circles.

By defaut, inner radius is of 20 pixels and outer is of 30.

It is possible to change these values in the setting box.



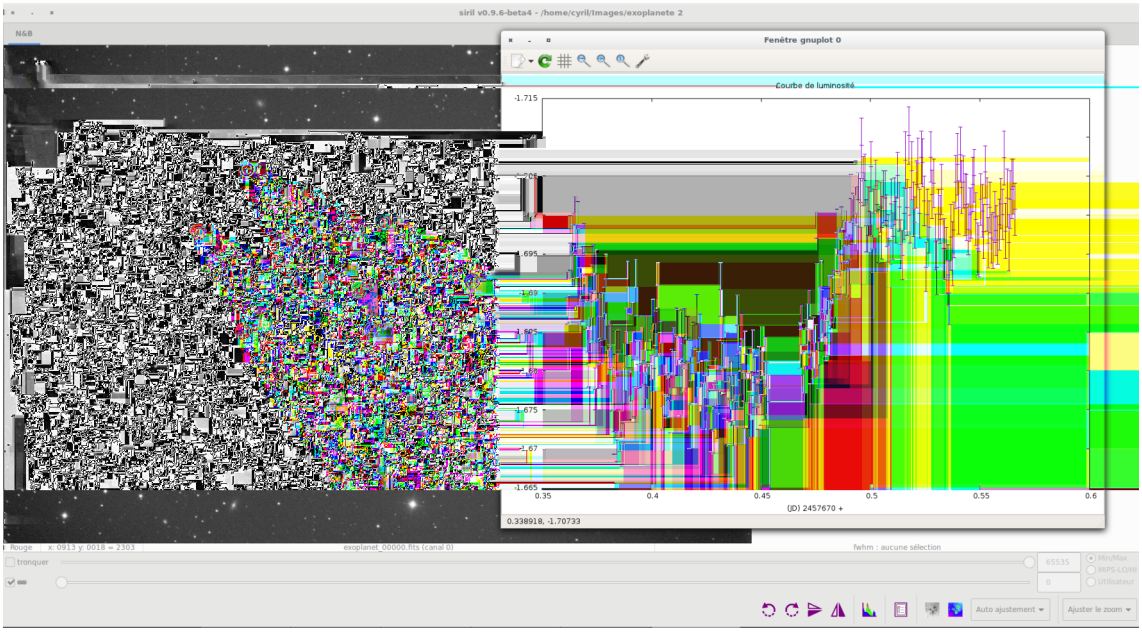
The other parameter is the conversion parameter of electrons by ADU[p.183] to have the real ADU. If not available, **2.3** should be sufficient.

Enter focal and pixels's size in parameters / camera & optics, Siril works in arc second. This updates the circles.

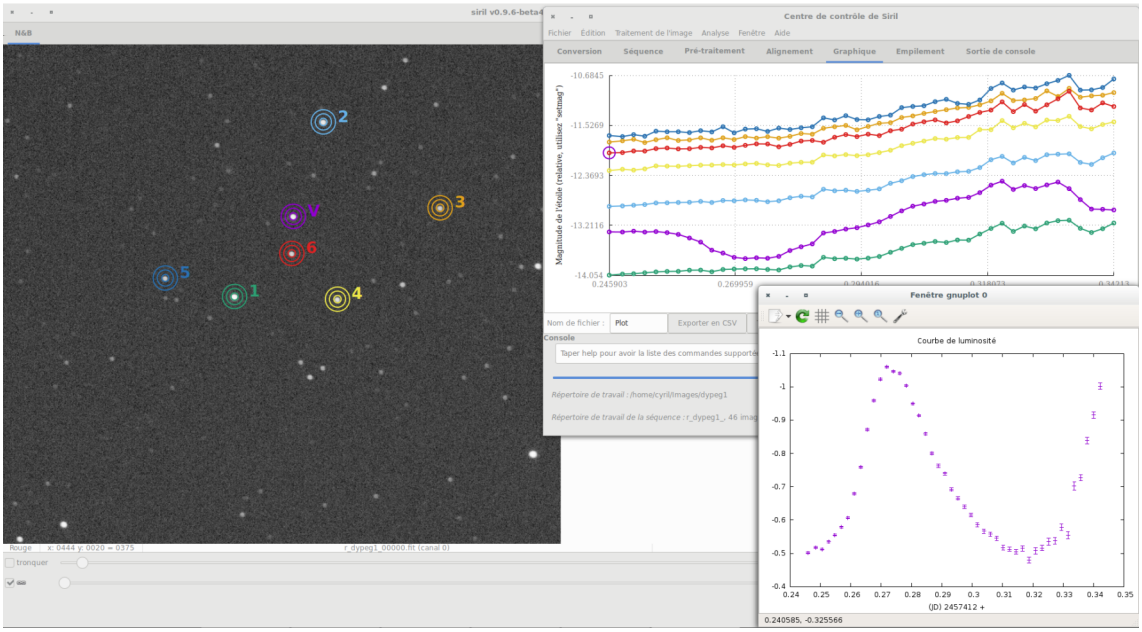
For our exoplanet images, this coefficient is stored in the header and its value is of 5.0.

This allows you to process your Variable Star or Planetary Transits sequences.

Planetary Transits.



Variable Star



See here [p.154]

1.6. Stacking

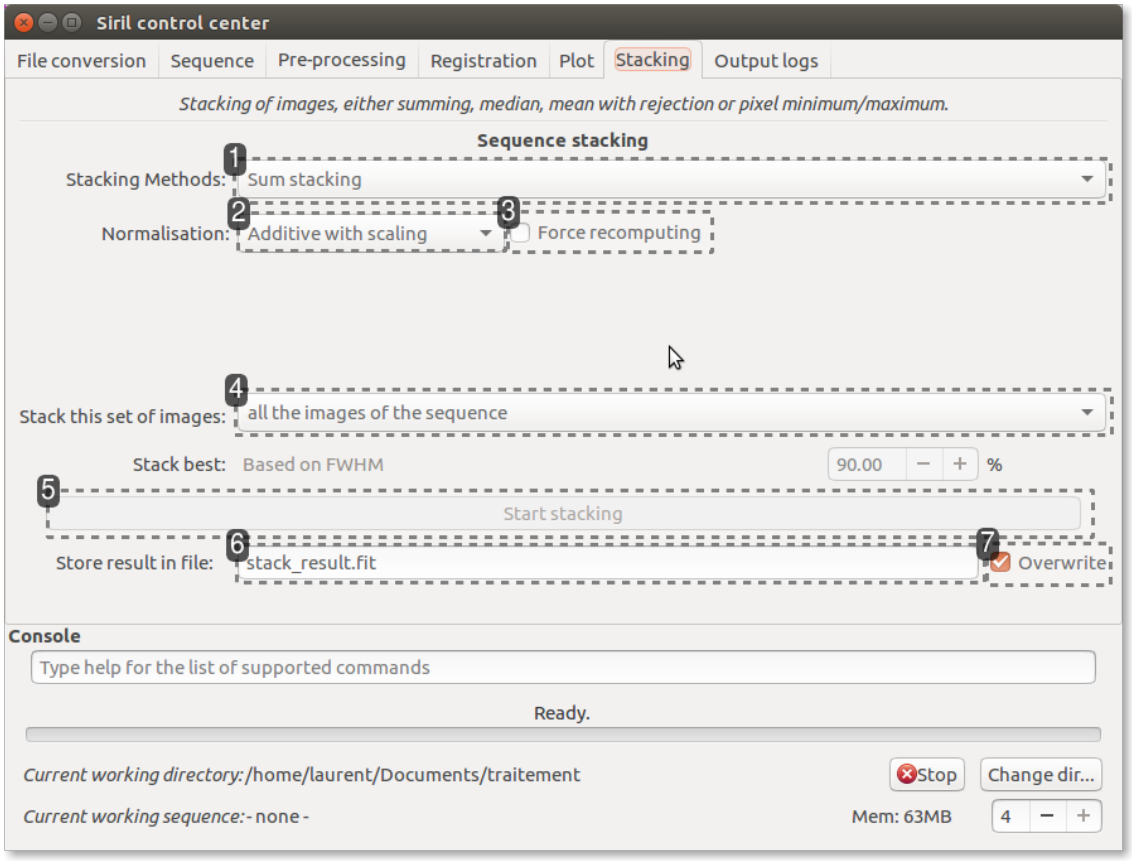
The final step to do with Siril is to stack the images. Go to the "stacking" tab, indicate if you want to stack all images, only selected images or the best images regarding the value of FWHM[p.189] previously computed. **Siril** proposes several algorithms for stacking computation.

Siril uses a 32-bit internal image to process the stacking, which is then saved to a 16-bit image.

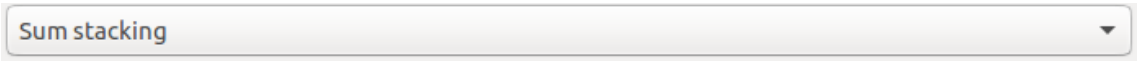
Stacking a planetary image

Siril provide an option to automatically align RGB channels [p.173] at the end of the stack.

Stacking

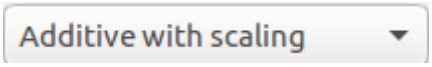


1 Stacking Methods



- Sum stacking
- Average stacking with rejection
- Median stacking
- Pixel maximum stacking
- Pixel minimum stacking

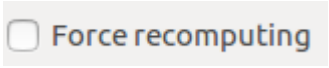
2 Normalization



- No normalization

- Additive
- Multiplicative
- Additive with scaling
- Multiplicative with scaling

3 Force recomputing

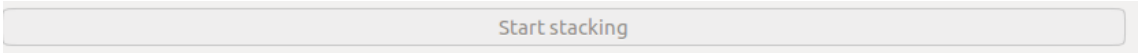


By default, Siril takes the normalization coefficients from seq file if already computed. However you can force Siril to recompute it before stacking.

4 Stack this set of images

- all the images of the sequence
- images previously manually selected from the sequence
- best images, automatically selected from the registration data (PSF)
- best images, automatically selected from the registration data (Planetary)

5 Start stacking

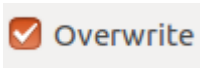


Start stacking the current sequence of images, using the method and other information provided above.

6 Store result in file:

Choose a file name

7 Overwrite



Overwrite file without notice if it already exists.

1.6.1. Stacking Methods

1.6.1.1. Sum Stacking

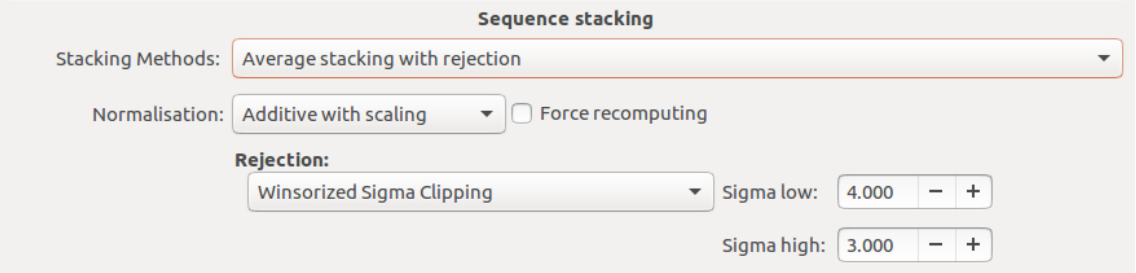
Sum Stacking

This is the simplest algorithm: each pixel in the stack is summed using 32-bit precision, and the result is normalized to 16-bit. The increase in signal-to-noise ratio (SNR) is proportional to $\sqrt{\mathbf{N}}$, where **N** is the number of images. Because of the lack of normalization, this method should only be used for **planetary processing**.

Stacking by sum has been parallelized to speed up deep sky lucky imaging file processing.

1.6.1.2. Average Stacking With Rejection

This method is used to reject deviant pixels iteratively.



Five rejection methods have been implemented:

Percentile Clipping

This is a one step rejection algorithm ideal for small sets of data (up to 6 images).

Sigma Clipping

This is an iterative algorithm which will reject pixels whose distance from median will be farthest than two given values in sigma units(σ_{low}, σ_{high}).

Median Sigma Clipping

This is the same algorithm than *`Sigma Clipping`* except than the rejected pixels are replaced by the median value of the stack.

Winsorized Sigma Clipping

This is very similar to Sigma Clipping method Sigma Clipping

It uses an algorithm based on Huber's work 1[p.200] 2[p.200]

Linear Fit Clipping

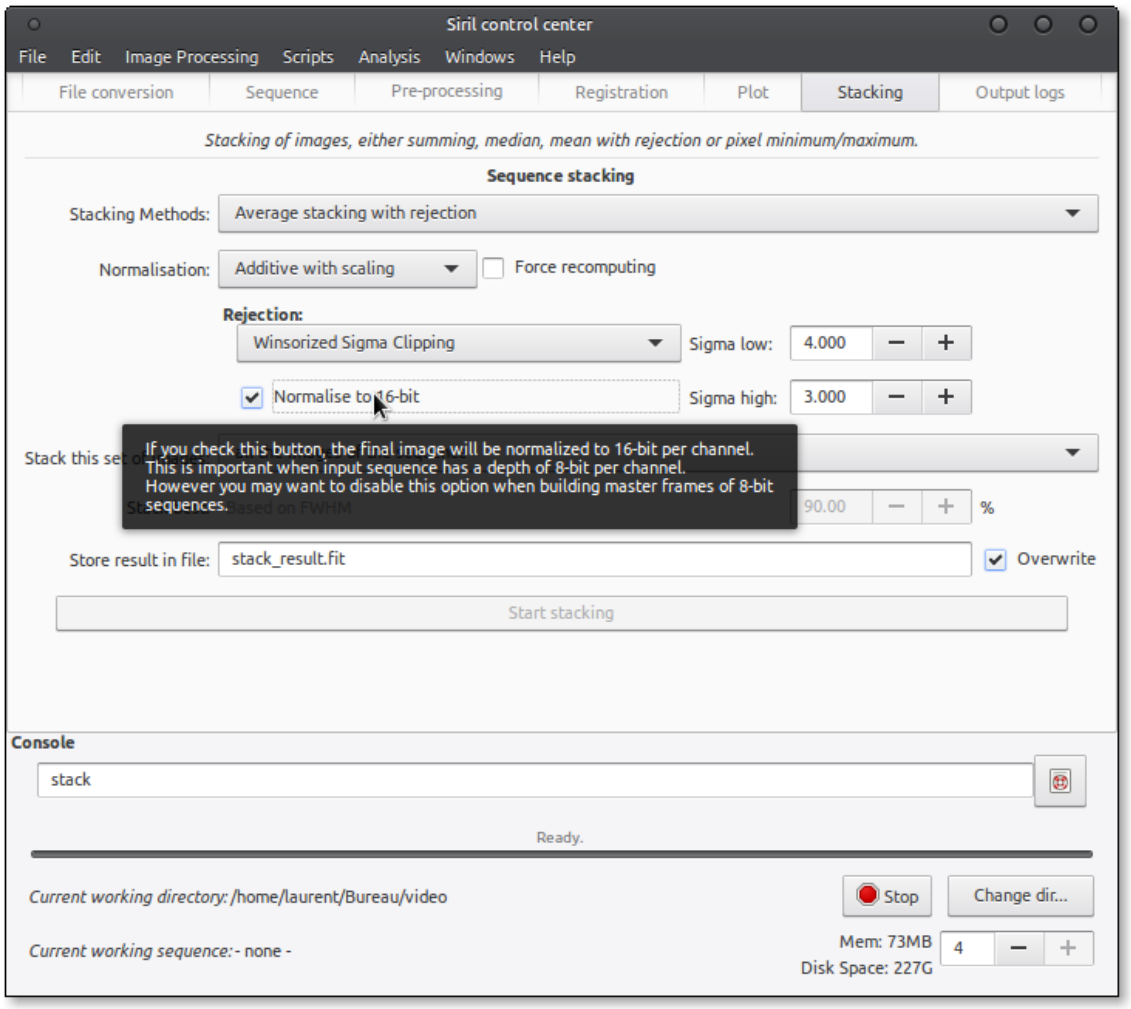
It fits the best straight line (y=ax+b) of the pixel stack and rejects outliers. This algorithm performs very well with large stacks and images containing sky gradients with differing spatial distributions and orientations.

This is an algorithm developed by Juan Conejero[p.200], main developer of PixInsight.

This algorithm is very efficient to remove satellite/plane tracks.

After pixel rejection, the mean of the stack is computed.

Normalization


Siril control center
File Edit Image Processing Scripts Analysis Windows Help
File conversion Sequence Pre-processing Registration Plot Stacking Output logs
Stacking of images, either summing, median, mean with rejection or pixel minimum/maximum.
Sequence stacking
Stacking Methods: Average stacking with rejection
Normalisation: Additive with scaling
Force recomputing
Rejection:
Winsorized Sigma Clipping
Sigma low: 4.000
Normalise to 16-bit
Sigma high: 3.000
If you check this button, the final image will be normalized to 16-bit per channel.
This is important when input sequence has a depth of 8-bit per channel.
However you may want to disable this option when building master frames of 8-bit sequences.
90.00 %
Store result in file: stack_result.fit
Overwrite
Start stacking
Console
stack
Ready.
Current working directory: /home/laurent/Bureau/video
Stop
Change dir...
Current working sequence: - none -
Mem: 73MB
Disk Space: 227G
4

8-bit image processing

It's possible to normalise at 16 bits per channel 8-bit images per channel.

Remember to disable this option during preprocessing, to build your Master Dark, Flats and Bias, and to reactivate it to make the final stack.

1.6.1.3. Median Stacking

Median Stacking

This method is mostly used for **dark/flat/offset stacking**. The median value of the pixels in the stack is computed for each pixel. As this method should only be used for dark/flat/offset stacking, it does not take into account shifts computed during registration. The increase in SNR is proportional to **0.8√N**.

1.6.1.4. Pixel Maximum Stacking

Pixel Maximum Stacking

This algorithm is mainly used to construct long exposure star-trails images. Pixels of the image are replaced by pixels at the same coordinates if intensity is greater.

1.6.1.5. Pixel Minimum Stacking

Pixel Minimum Stacking

This algorithm is mainly used for cropping sequence by removing black borders. Pixels of the image are replaced by pixels at the same coordinates if intensity is lower.

1.6.2. Normalization

If one of these items is selected, a normalization process will be applied to all input images before stacking.

Normalization matches the mean background of all input images, then, the normalization is processed by multiplication or addition.

Keep in mind that both processes generally lead to similar results but multiplicative normalization is preferred for image which will be used for multiplication or division as flat field.

Scale matches dispersion by weighting all input images.

This tends to improve the signal-to-noise ratio and therefore this is the option used by default with the additive normalization.

> The offset and dark masters should not be processed with normalization. However, multiplicative normalization must be used with flat-field frames.

Possible values

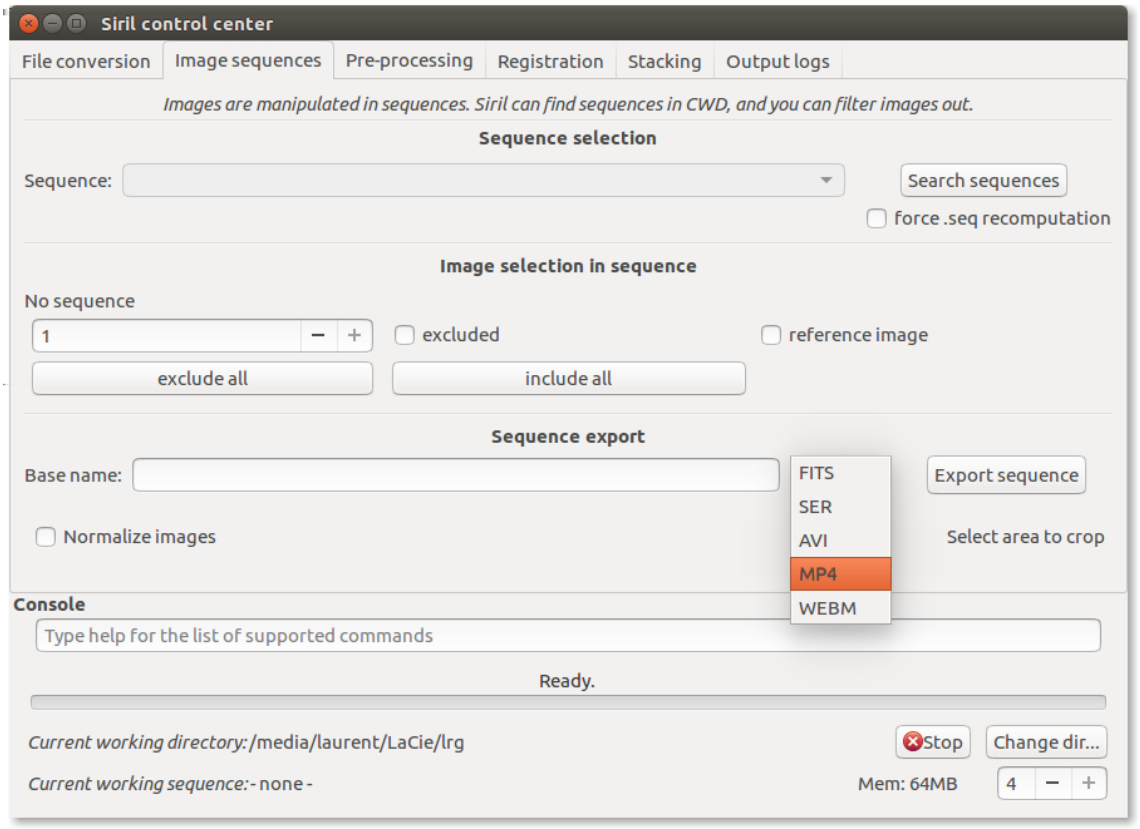
- No normalization
- Additive
- Multiplicative
- Additive with scaling
- Multiplicative with scaling

1.7. Sequence export

Siril can export sequences in the following formats:

- FITS
- SER
- AVI
- MP4
- WEBM

In Images sequences tab

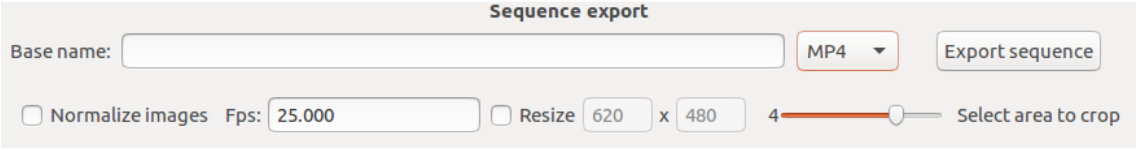


★ Supported export options

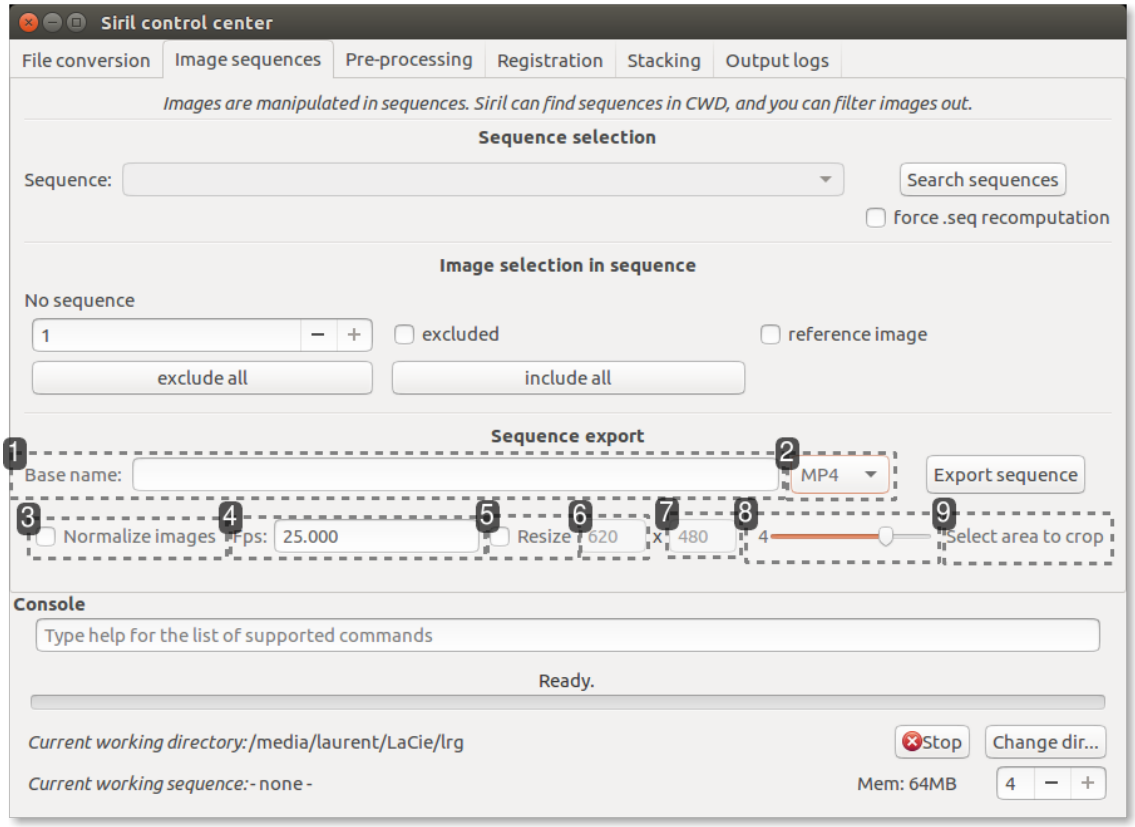
FITS and SER

AVI

MP4 and WEBM



All sequence export options

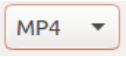


1

Base name

Base name for the exported sequence

2



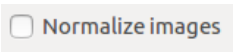
Export sequence format.

- FITS
- SER
- AVI
- MP4
- WEBM

MP4 and WEBM replaces GIF for modern Web publication

AVI is not compressed.

3



Normalize images

Activate image normalisation for the export.

This will make all images the same brightness and is thus recommended for sequences with inconsistent exposures or pixel values.

> A first computation pass will be done on all images before exporting, it may take some time, only the first export of a sequence.

4

Frame rate

In frames per second

5

Resize

Check this box if you want to resize exported sequence. Note that this will degrade image quality.

6

Width of output video

In pixels

7

Height of output video

In pixels

8

Film quality:

Lower value will use more compression and less bit-rate, also resulting is smaller file size.

9

Select area to crop

You can crop and normalize sequence images

1.8. Available commands

Reference list of available commands.

This manual is a summary of the different commands you can find in Siril.

Automatic processing

With a text file

It's possible to perform automatic processing

Write command in a file, one command per line

save the file *`filename`*

then load it in the console with `@filename`

The <SPACE> character is the delimiter between arguments. If you need to have spaces inside the arguments, you can use the quote or double quote, just like in a shell.

With a script

Since version 0.9.9 you can create your custom scripts.

For their use see the scripting section.

Sample script for preprocessing DSLR images

```
############################################
# Script for Siril 0.9.9
# April 2018
# (C) Cyril Richard
```

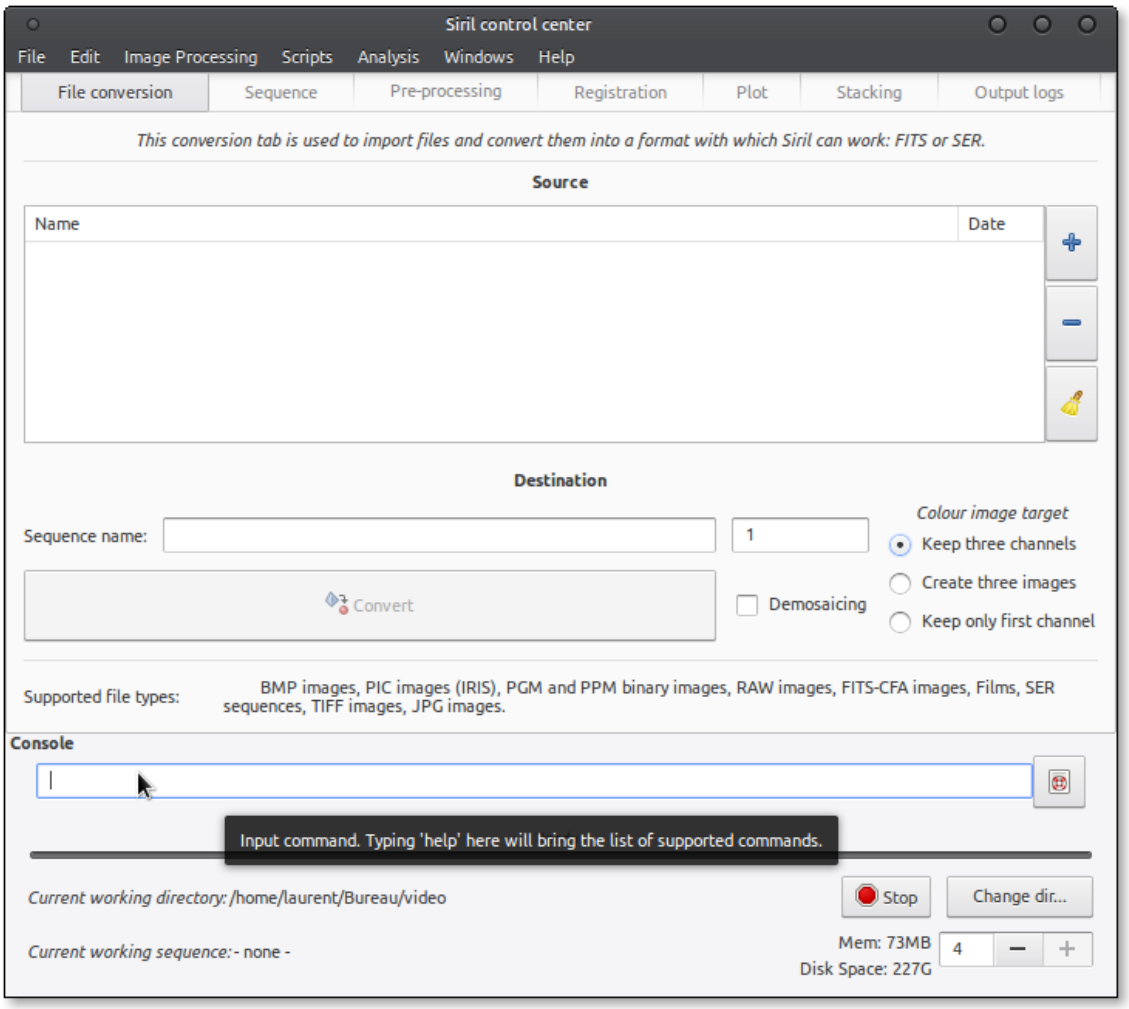
```
# DSLR_preprocessing v1.1
########### PREPROCESSING SCRIPT ###########
# Script for DSLR color camera preprocessing
# needs 4 sets of RAW images in the working
# directory, within 4 directories:
# biases, flats, darks and lights
#
############################################
#build master-bias
cd biases
convertraw bias_
stack bias_ rej 3 3 -nonorm
cd ..
cd flats
#preprocess flats
convertraw flat_
preprocess flat_ -bias=../biases/bias_stacked.fit
#build master-flat
stack pp_flat_ rej 3 3 -norm=mul
cd ..
#build master-dark
cd darks
convertraw dark_
stack dark_ rej 3 3 -nonorm
cd ..
#preprocess lights
cd lights
convertraw light_
preprocess light_ -dark=../darks/dark_stacked.fit -flat=../flats/pp_flat_stacked.fit -cfa -debayer
#align lights
register pp_light_
#stack calibrated lights
stack r_pp_light_ rej 3 3 -norm=addscale
#That's a workaround here to save result in another place.
#These lines are not mandatory
#load image in memory
load r_pp_light_stacked.fit
#Compute noise
bgnoise
```

#and save it at root
cd ..
save result.fit
close

Download the latest version

https://free-astro.org/svn/siril/branches/0.9/scripts/DSLR_preprocessing.ssf

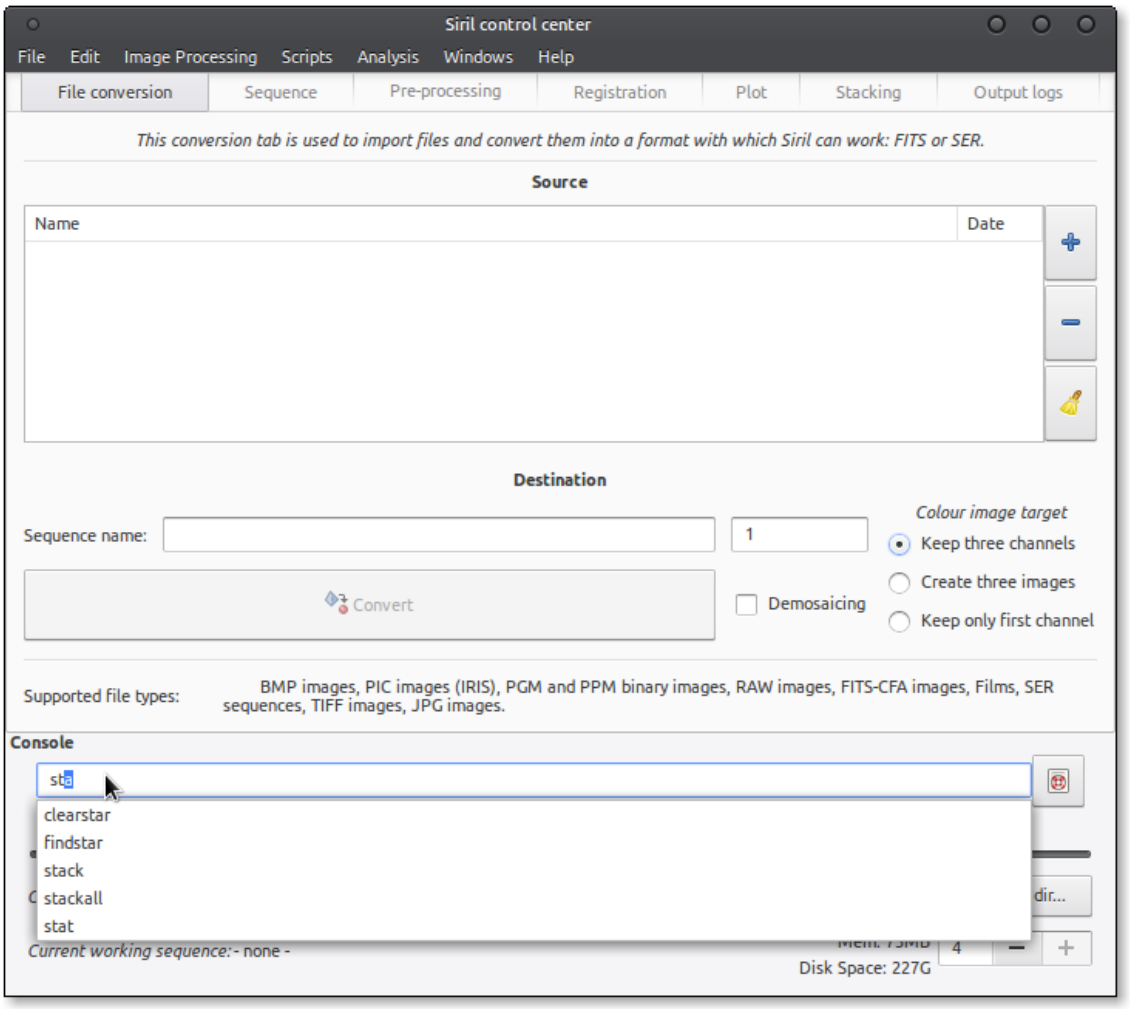
You can use the commands directly in the console


Siril control center
File Edit Image Processing Scripts Analysis Windows Help
File conversion Sequence Pre-processing Registration Plot Stacking Output logs
This conversion tab is used to import files and convert them into a format with which Siril can work: FITS or SER.
Source
Name Date
Destination
Sequence name:
1
Colour image target
Keep three channels
Create three images
Keep only first channel
Convert
Demosaicing
Supported file types:
BMP images, PIC images (IRIS), PGM and PPM binary images, RAW images, FITS-CFA images, Films, SER sequences, TIFF images, JPG images.
Console
Input command. Typing 'help' here will bring the list of supported commands.
Current working directory: /home/laurent/Bureau/video
Stop
Change dir...
Current working sequence: - none -
Mem: 73MB
Disk Space: 227G
4

The `help` command directly explains the syntax to use.

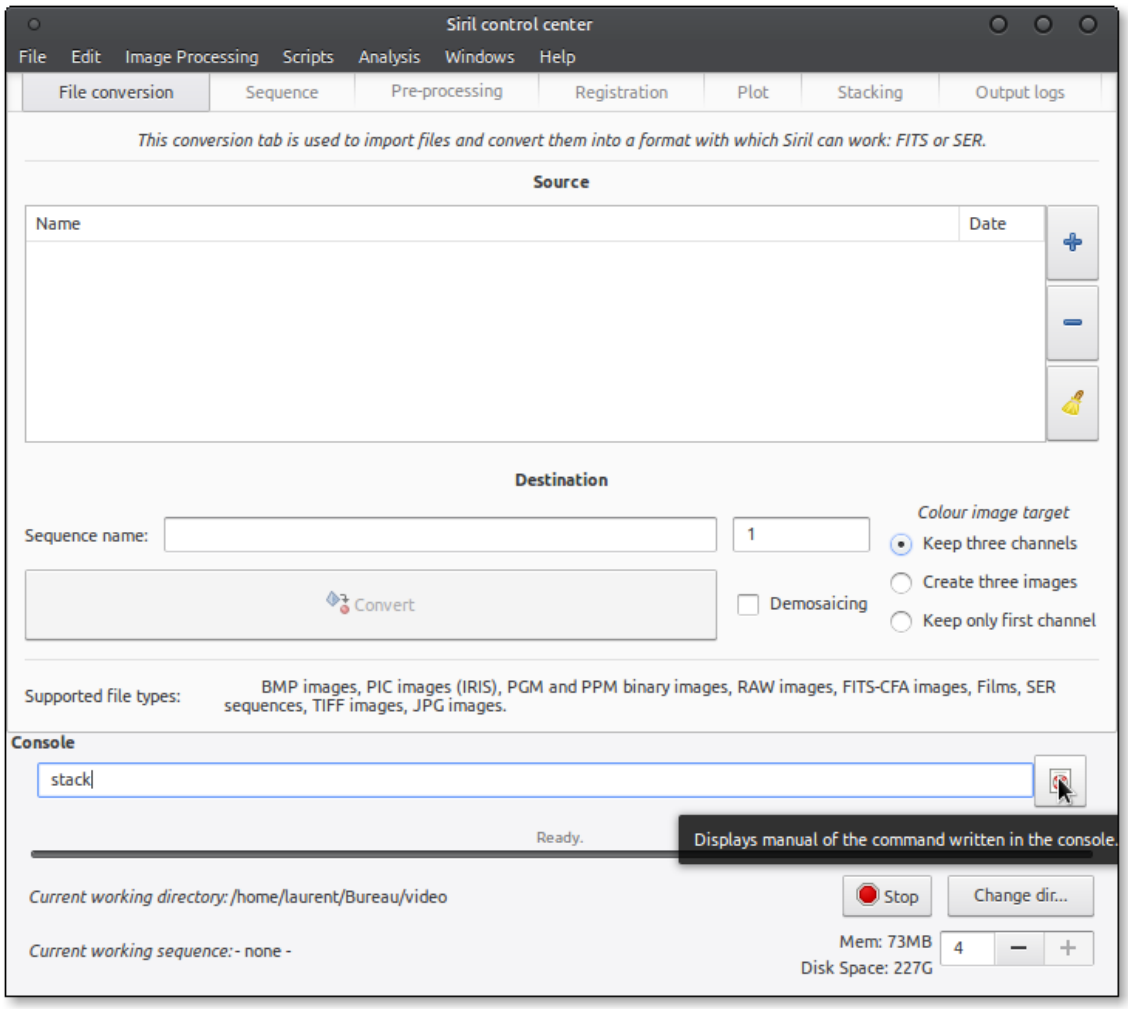
Since version 0.9.9 self-completeness is active in the console

When you enter a command in the console, an order list is proposed according to your input:

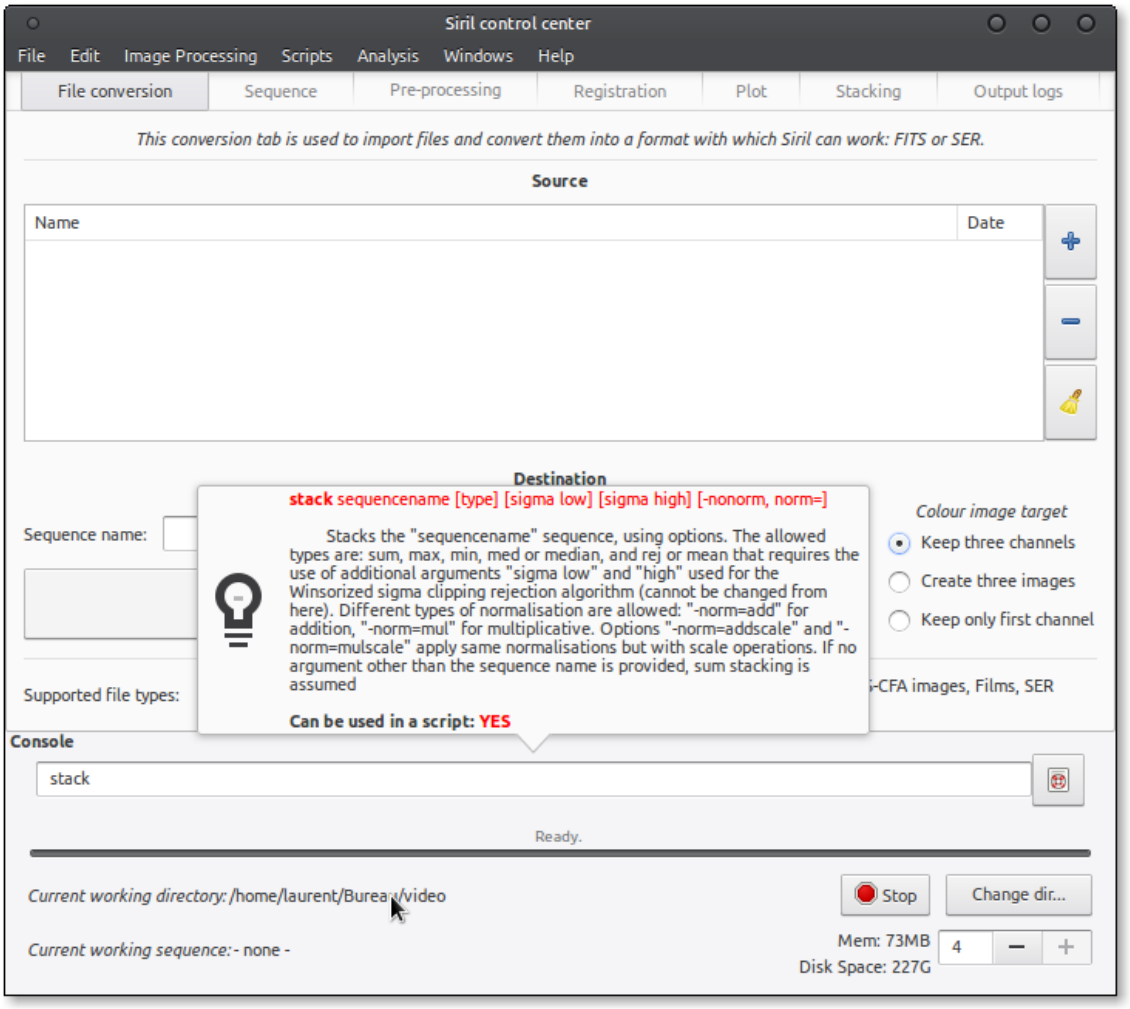


The instruction command manual

By clicking on the icon to the right of the console, you access the instruction commands manual:


Siril control center
File Edit Image Processing Scripts Analysis Windows Help
File conversion Sequence Pre-processing Registration Plot Stacking Output logs
This conversion tab is used to import files and convert them into a format with which Siril can work: FITS or SER.
Source
Name Date
Destination
Sequence name:
1
Convert
Demosaicing
Colour image target
Keep three channels
Create three images
Keep only first channel
Supported file types: BMP images, PIC images (IRIS), PGM and PPM binary images, RAW images, FITS-CFA images, Films, SER sequences, TIFF images, JPG images.
Console
stack
Ready.
Displays manual of the command written in the console.
Current working directory: /home/laurent/Bureau/video
Stop
Change dir...
Current working sequence: - none -
Mem: 73MB
Disk Space: 227G
4

Manual entry for `Stack`


Siril control center
File Edit Image Processing Scripts Analysis Windows Help
File conversion Sequence Pre-processing Registration Plot Stacking Output logs
This conversion tab is used to import files and convert them into a format with which Siril can work: FITS or SER.
Source
Name Date
Destination
Sequence name:
stack sequencename [type] [sigma low] [sigma high] [-nonorm, norm=]
Stacks the "sequencename" sequence, using options. The allowed types are: sum, max, min, med or median, and rej or mean that requires the use of additional arguments "sigma low" and "high" used for the Winsorized sigma clipping rejection algorithm (cannot be changed from here). Different types of normalisation are allowed: "-norm=add" for addition, "-norm=mul" for multiplicative. Options "-norm=addscale" and "-norm=mulscale" apply same normalisations but with scale operations. If no argument other than the sequence name is provided, sum stacking is assumed
Can be used in a script: YES
Colour image target
Keep three channels
Create three images
Keep only first channel
Supported file types:
-CFA images, Films, SER
Console
stack
Ready.
Current working directory: /home/laurent/Bureau/video
Stop
Change dir...
Current working sequence: - none -
Mem: 73MB
Disk Space: 227G
4

Commands list

You will find below the list of the commands and a description of their use.

addmax

`addmax filename`

addmax compute a new image IMG with IMG_1 and IMG_2. The pixel of IMG_1 is replaced by the pixel at the same coordinates of IMG_2 if the intensity of 2 is greater than 1.

> Do not forget to save the result.

> Can not be used in a script.

bg

`bg`

Returns the background level of the image loaded in memory.

Can be used in a script

bgnoise

`bgnoise`

Returns the background noise level.

Can be used in a script

cd

`cd directory`

Defined the new working *`directory`*

Set the new current working directory. directory can contain the ~ token, expanded as the home directory, directories with spaces in the name can be protected using single or double quotes.

Examples:

- `cd ~/M42`
- cd `'../OIII 2x2/'`

Can be used in a script

cdg

`cdg`

Return the coordinates of the center of gravity of the image.

Can be used in a script

clear

`clear`

Clears the graphical output logs.

clearstar

`clearstar`

Clear all the stars saved in memory and displayed on the screen.

Can not be used in a script.

convertraw

`convertraw basename [-debayer]`

Convert DSLR RAW into Siril's FITS images.

The argument `basename` is the basename of the new sequence.

The option `-debayer` applies demosaicing to images

Can be used in a script

close

`close`

Closes the open image and the open sequence correctly, if necessary.

Can be used in a script

cosme

`cosme filename`

Apply the local mean to a set of pixels on the in-memory image (cosmetic correction). The coordinate of this pixels are in an ASCII file [list file]. COSME is adapted to correct residual hot and cold pixels after preprocessing.

The line P x y type will fix the pixel at coordinates (x, y) type is an optional character (C or H) specifying to Siril if the current pixel is cold or hot. This line is created by the command find_hot but

you also can add some lines manually.

The line C x 0 type will fix the bad column at coordinates x.

The line L y 0 type will fix the bad line at coordinates y.

Can be used in a script

cosme_cfa

`cosme_cfa filename`

Same function that COSME but applying to *`RAW CFA images.`*

Can be used in a script

crop

`crop x, y, width, height`

Crops the current image within the rectangle previously selected.

Can be used in a script

ddp

`ddp level coef sigma`

Performs a DDP (digital development processing) as described first by Kunihiko Okano*[https://www.sbig.com/astronomy/hall-of-fame/kunihiko-okano/]*.

This implementation is the one described in IRIS. It combines a linear distribution on low levels (below level) and a non-linear on high levels. It uses a Gaussian filter of sigma sigma multiplies the resulting image by coef. The typical values for sigma are included between 0.7 and 2

Can not be used in a script.

entropy

`entropy`

Computes the entropy of the opened image on the displayed layer, only in the selected area if one has been selected or in the whole image else.

The entropy is one way of measuring the noise or the details in an image.

Can be used in a script

exit

`exit`

Quits the application.

Can be used in a script

extract

`extract NbPlane`

Extracts NbPlane Planes of Wavelet domain.

Can be used in a script

fdiv

`fdiv filename scalar`

Divides the image in memory by the image given in argument. The resulting image is multiplied by the value of the scalar argument. Please check that the image is in the working directory

Check that the image is in the working directory.

See also idiv

Can be used in a script

fftd

`fftd modulus phase`

Applies a Fast Fourier Transform to the image loaded in memory. Modulus and phase given in argument are saved in FITS files.

Can be used in a script

ffti

`ffti modulus phase`

This function is used to retrieve corrected image applying an inverse transformation. The modulus and phase used are the files given in argument.

Can be used in a script

fill

`fill value x y width height`

Fills the whole current image (or selection) with pixels having the value intensity.

Can be used in a script

fill2

`fill2 value x y width height`

Same command than fill but this is a symmetric fill of a region defined by the mouse. Used to process an image in the Fourier (FFT) domain.

Can be used in a script

find_cosme

`find_cosme cold_sigma hot_sigma`

Applies an automatic detection of cold and hot pixels following the thresholds written arguments

Can be used in a script

find_cosme_cfa

`find_cosme_cfa cold_sigma hot_sigma`

Same command than `find_cosme` but for monochromatic CFA images.

Can be used in a script

find_hot

`find_hot filename cold_sigma hot_sigma`

The command provides a file lists "filename" (format text) in the working directory which contains the coordinates of the pixels which have an intensity "hot_sigma" times higher and "cold_sigma" lower than standard deviation. We generally use this command on a master-dark file.

Can be used in a script

findstar

`findstar`

Detects stars having a level greater than a threeshold computed by Siril.

> After that, a dynamic PSF is applied and Siril rejects all detected structures that don't fulfill a set of prescribed detection criteria. Finally, a circle is drawn around detected stars. See also the command clearstar.

The algorithm is based on the publication of Mighell, K. J. 1999, in ASP Conf. Ser., Vol. 172, Astronomical Data Analysis Software and Systems VIII, eds. D. M. Mehringer, R. L. Plante, & D. A. Roberts (San Francisco: ASP), 317.

> Can not be used in a script.

fmedian

`fmedian ksize modulation`

Performs a median filter of size ksize ×ksize (ksize MUST be odd) to the original image with a modulation parameter modulation. The output pixel is computed as : out =mod × m+(1−mod)×in, where m is the median-filtered pixel value. A modulation's value of 1 will apply no modulation.

> **Can be used in a script**

fmul

`fmul scalar`

Multiplies the loaded image by the `scalar` given in argument.

> **Can be used in a script**

fixbanding

`fixbanding amount sigma`

Try to remove the canon banding. "Amount" define the amouont of correction. Sigma defines a protection level of the algorithm, higher sigma gives higher protection.

> **Can be used in a script**

gauss

`gauss sigma`

Performs a Gaussian filter with the given sigma.

Can be used in a script

help

`help`

Gives the available commands in output logs tab.

Can not be used in a script.

histo

`histo layer`

Calculates the histogram of the image layer in memory and produces file histo_[layer name].dat in the working directory.

layer = 0, 1 or 2 with 0=red, 1=green and 2=blue.

Can be used in a script

idiv

`idiv filename`

Divides the image into memory by the image given in argument.

Divides the image in memory by the image given in argument. Please check that the image is in the working directory.

See also `fdiv`.

Can not be used in a script.

imul

`imul filename`

Multiplies the image in memory by the image given in argument.

Please check that the image is in the working directory.

Can not be used in a script.

isub

`isub filename`

Substracts the image in memory by the image given in argument.

Please check that the image is in the working directory.

Can not be used in a script.

load

`load filename`

`load filename.ext`

Loads the fits image filename; it first attempts to load filename, then filename.fit and finally filename.fits and after, all supported format, aborting if none of these are found. These scheme is applicable to every Siril command implying reading files. Fits headers MIPS-HI and MIPS-LO are read and their values given to the current viewing levels.

Writing a known extension at the end of filename will load the image filename.ext: this is used when numerous files have the same name but not the same extension.

Extensions supported are :

- .fit, *.fits, *.fts
- .bmp / *.jpg, *.jpeg / *.png / *.tif, *.tiff
- .ppm, *.pgm
- .pic (IRIS file)

Can be used in a script

log

`log`

Computes and applies a logarithmic scale to the current image.

Can be used in a script

ls

`ls`

This command lists files and directories in the working directory.

Can not be used in a script.

mirrorx

`mirrorx`

Rotates the image around a vertical axis.

Can be used in a script

mirrory

`mirrory`

Rotates the image around an horizontal axis.

Can be used in a script

new

`new width height nb_layers`

Creates a new image filled with zeros with a size of width x height. The image is in 16-bit format, and it contains nb_layers layers, nb_layers being 1 or 3.

It is not saved, but displayed and can be saved afterwards.

Can not be used in a script.

nozero

`nozero level`

Replaces null values by level values. Useful before an `idiv` ou `fdiv` operation.

Can be used in a script

offset

`offset value`

Adds the constant value to the current image. This constant can take a negative value. As Siril uses unsigned fit files, if the intensity of the pixel become negative its value is replaced by 0 and by 65535 (for a 16-bit file) if the pixel intensity overflows.

> To check the minimum and maximum intensities values, click on the Auto level button and note the low and high threshold.

> **Can be used in a script**

preprocess

`preprocess sequencename [-bias=, -dark=, -flat=] [-cfa] [-debayer] [-flip]`

Preprocess the sequence `sequencename` using `bias`, `dark` and `flat` given in argument. It's possible to specify if images are CFA for cosmetic correction purposes with the option `-cfa` and also to demoisaic images at the end of the process with `-debayer`.

The -`flip` options tells to Siril to read image from up to bottom for demosaicing operation.

> **Can be used in a script**

psf

`psf`

Make a selection around a star and call the command PSF. It will give you:

- The centroid coordinates (x0 and y0) in pixel units, which is the position of the center of symmetry of the fitted PSF.
- The FWHM on the X and Y axis.
- The rotation angle of the X axis with respect to the centroïde*[https://www.oqlf.gouv.qc.ca/ressources/bibliotheque/dictionnaires/terminologie_geomatique/centroide.html]* coordinates.
- The average local background.
- The maximal intensity of the star: this is the peak value of the fitted function, located at the centroid coordinates x0 and y0.
- The relative magnitude of the star.
- The RMSE. This is an estimate of fitting quality. The smaller the RMSE is, the better the function is fitted.

To be relevant, the selection MUST be done on a non-saturated star.

Can not be used in a script.

register

`register sequence [-norot] [-drizzle]`

Effectue des transformations géométriques sur les images de la séquence donnée en l'argument afin qu'elles puissent être superposée à l'image de référence. Le nom de la séquence de sortie commence par le préfixe `r_`. Utilisant des étoiles pour l'alignement, cet algorithme ne fonctionne qu'avec des images de ciel profond. L'option `-norot` effectue une translation seule avec aucune création de nouvelle séquence alors que l'option `-drizzle` applique un drizzle x2 sur les images.

Can be used in a script

resample

`resample factor`

Resample image with a factor *`factor`*

Can be used in a script

rl

`rl iterations sigma`

Restore an image uing the Richardson-Lucy method. `iterations` is the number of iterations to be performed (typically between 10 and 50). `sigma` is the size of the kernel to be applied.

Can be used in a script

rmgreen

`rmgreen type`

rmgreen is a chromatic noise reduction filter. It removes green noise in the current image. This filter is based on PixInsight's SCNR Average Neutral algorithm and it is the same filter used by HLVG plugin in Photoshop. In command line, the lightness is always preserved. For image processing without L* preservation use the graphical tool box and uncheck the corresponding button.

`Type`=1 stands for Average Neutral Protection, while type=2 stands for Maximum Neutral

> Protection.

Can be used in a script

rotate

`rotate degree`

Rotates the image of an angle of degree value.

> This function requieres the installation of libopencv.

Can be used in a script

rotatepi

`rotatepi`

Rotates the image of an angle of 180° around its center.

> This is equivalent to the command `rotate180` or `rotate -180.`

Can be used in a script

satu

`satu coeff`

Enhance the global saturation of the image.

> Try iteratively to obtain best results.

> For exemple: `Satu 0,1`

Can be used in a script

save

`save filename`

Save current image to filename.fit. Fits headers MIPS-HI and MIPS-LO are added with values corresponding to the current viewing levels..

Can be used in a script

savebmp

`savebmp filename`

Save current image under the form of a bitmap file with 8bits per channel: filename.bmp (BMP 24 bits).

This function is like a screenshot of what you see with the levels updated. This is very usefull to share an image in the bitmap format.

Can be used in a script

savejpg

`savejpg filename [quality]`

Save current image into a JPG file. You have the possibility to adjust the quality of the compression. A value 100 for quality parameter offers best fidelity while a low value increases the compression ratio. If no value is specified, it holds a value of 100.

This function is like a screenshot of what you see with the levels updated.

If no value is specified, it holds a value of 100.

This command is very useful to share an image in the jpeg format on the forums for example.

Can be used in a script

savepng

`savepng filename [quality]`

Save current image into a png file.

Can be used in a script

savepnm

`savepnm filename`

Save current image under the form of a Netpbm file format with 16bits per channel. The extension of the output will be filename.ppm for RGB image and filename.pgm for gray-level image.

> More details about the Netpbm format here[http://en.wikipedia.org/wiki/Netpbm_format]

Can be used in a script

savetif

`savetif filename`

Save current image under the form of a uncompressed TIFF file with 16bits per channel.

Can be used in a script

savetif8

`savetif8 filename`

Same command than savetif but the output file is saved in 8bits per channel.

Can be used in a script

select

`select from to`

This command allows easy mass selection of images in the loaded sequence (from - to, to included).

> `select 0 0`
> selects the first.
>
> `select 1000 1200`
> selects 201 images starting from number 1000

> The second number can be greater than the number of images to just go up to the end.

Can not be used in a script.

seqcrop

`seqcrop`

Crops the loaded sequence.

Can not be used in a script.

seqfind_cosme

`seqfind_cosme sequencename cold_sigma hot_sigma`

Same command than `find_cosme` but for the sequence `sequencename`.

Can be used in a script

seqfind_cosme_cfa

`seqfind_cosme_cfa sequencename cold_sigma hot_sigma`

Same command than `find_cosme_cfa` but for the sequence `sequencename`.

seqpsf

`seqpsf`

Same command than `PSF` but for the sequence.

Result are dumped in the console in a form that can be used to produce brightness variation curves.

Can not be used in a script.

setcpu

`setcpu number`

Defines the number of processing threads used for calculation. Can be as high as the number of virtual threads existing on the system, which is the number of CPU cores or twice this number if hyperthreading (Intel HT) is available.

Can be used in a script

setext

`setext extension`

Sets the extension used and recognized by sequences. The argument "extension" can be "fit", "fts" or "fits".

Can be used in a script

setmag

`setmag magnitude`

Defines the magnitude constant by selecting a star and giving the true magnitude. All PSF computations will return the true magnitude after this command.

To reset the magnitude constant see `unsetmag`.

Can not be used in a script.

setmagseq

`setmagseq magnitude`

This command is only valid after having run seqpsf or its graphical counterpart (select the area around a star and launch the *`psf analysis`* for the sequence, it will appear in the graphs).

This command has the same goal as *`setmag`* but recomputes the reference magnitude for each image of the sequence where the reference star has been found.

When running the command, the last star that has been analyzed will be considered as the reference star.

Displaying the magnitude plot before typing the command makes it easy to understand.

To reset the reference star and magnitude offset, see *`unsetmagseq`*.

Can not be used in a script.

split

`split r g b`

The command splits the color image into three distincts files (one for each color) and save them in r g and b file.

Can not be used in a script.

stack

`stack sequencename [type] [sigma low] [sigma high] [-nonorm, norm=]`

Stack the sequence `sequencename`, using options.

The allowed types are: **sum**, **max**, **min**, **med** ,**median**, and **rej** or **mean** mean that requires the use of additional arguments `sigma low` and `sigma high` used for the *`Winsorized sigma clipping`* rejection algorithm (cannot be changed from here).

If no argument other than the sequence name is provided, sum stacking is assumed.

No image filtering of the sequence is applied: all are selected to be stacked.

Different types of normalisation are allowed: `-norm=add` for addition, `-norm=mu` for multiplicative.

Options `-norm=addscale` and `-norm=mulscale` apply same normalisations but with scale operations.

The stacked image sequence is created with the "_stacked" suffix and the configured FITS file extension.

This command has been added since version 0.9.9

Can be used in a script

stackall

`stackall [type] [sigma low] [sigma high] [-nonorm, norm=]`

Opens all sequences in the CWD and stacks them with the optionally specified stacking type or with

sum stacking. See STACK command for options description.

Can be used in a script

stat

`stat`

Returns global statistic of the current image. If a selection is made, the command returns global statistic within the selection.

Can be used in a script

threshlo, threshhi, thresh

These are threshold functions:

- `threshlo 40` replaces values below 40 with 40;
- `threshhi 1000` replaces values above 1000 with 1000;
- `thresh 40 1000` does both.

Can be used in a script

unselect

`unselect from to`

This command allows easy mass unselection of images in the loaded sequence (from - to).

See `select`.

Can not be used in a script.

unsetmag

`unsetmag`

Reset the magnitude constant to 0.

Voir `setmag`

Can not be used in a script.

unsetmagseq

Reset the magnitude calibration and reference star for the sequence.

Can not be used in a script.

unsharp

`unsharp sigma multi`

Applies to the working image an unsharp mask with sigma sigma and coefficient multi.

Can be used in a script

visu

`visu low high`

Displays an image with low and high as the low and high threshold.

Can not be used in a script.

wavelet

`wavelet plan_number type`

Computes the wavelet transform on plan_number plans using linear (type=1) or bspline (type=2) version of the 'a trous' algorithm. The result is stored in a file as a structure containing the planes, ready for weighted reconstruction with wrecons.

Can be used in a script

wrecons

`wrecons c1 c2 ... cn`

Reconstructs to current image from the planes previously computed with wavelet and weighted with coefficients c1, c2, ..., cn according to the number of planes used for wavelet transform

Can be used in a script

1.9. A powerful astronomy image viewer

A Display Mode viewer is used to improve the visibility of an image.

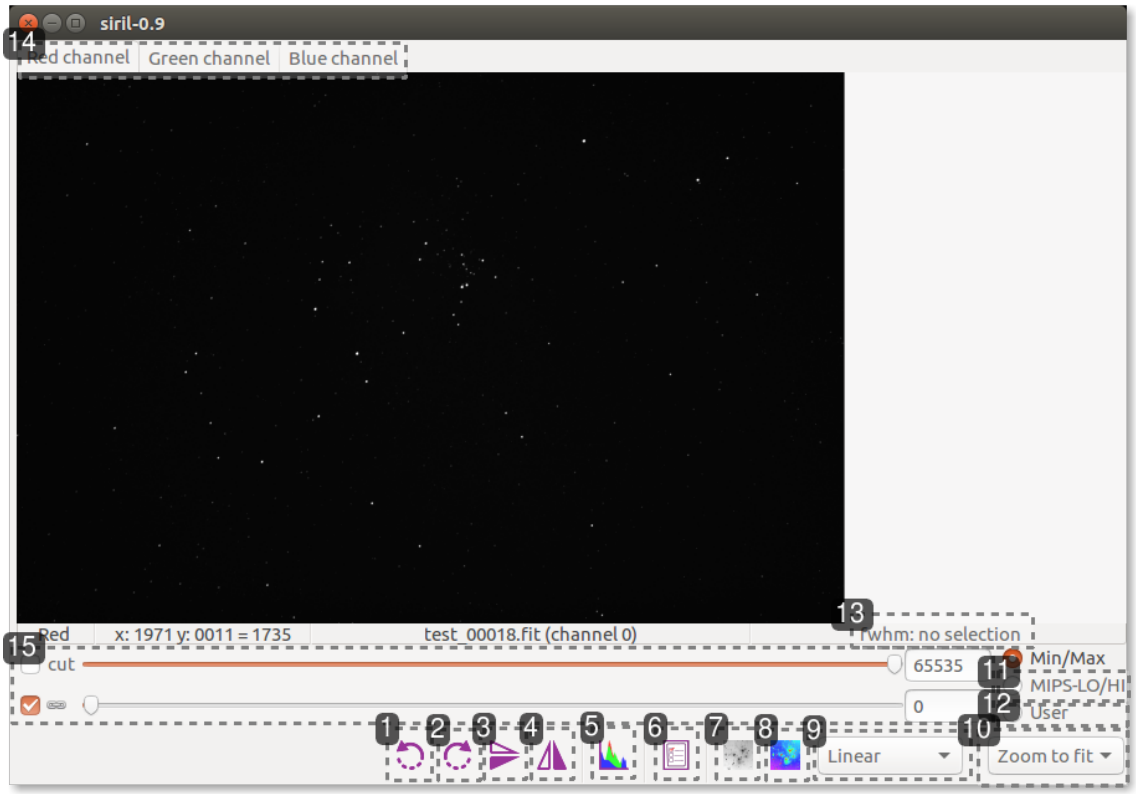
Without altering its pixel data in any way, similarly to what DS9[p.200] scale does.

- The grey window is used to display color components for the image,
- control their visual composition in the color image,
- and give information about the displayed frame,
- pixel value when moving the mouse on a pixel,

Affichage de la valeur du pixel

- file name and layer number,
  zoom value,
- FWHM [p.189] value of the selection if it applies,
- and some display tools.

The display modes allow the scaling of the source image file to the displayed image to be changed. sliders and display modes are used to improve the visibility of some details in an image, or more precisely to change the dynamics of the viewing, without altering pixel data from the image file in any way. This is only for display purposes.



1 Rotate

Rotate 90° counter clockwise

2 Rotate

Rotate 90° clockwise

3 Mirror

Horizontal mirror

4 Mirror

Vertical mirror

5 Histogram

Open / close ☞histogram toolbox [p.84]

6 List of images

Show/hide ☞list of images [p.78] in the sequence with registration data ☞ [p.84]

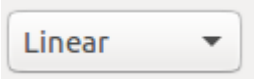
7 Negative view

☞Colors maps viewer [p.79] ☞ [p.84]

8 False color rendering

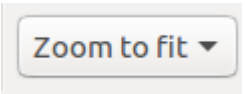
Switch to normal and rainbow color map ☞Colors maps viewer [p.79] ☞ [p.84]

9 Display modes



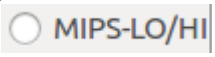
The different display modes

10 Zoom



Zoom level

11 MIPS-LO/HI

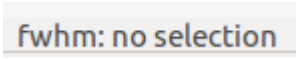


If this option is checked Fits headers MIPS-HI and MIPS-LO are read and their values given to the current viewing levels.

12 User

User is the values defined by the user. As soon as you touch the sliders, it uses user mode.

13 fwhm value

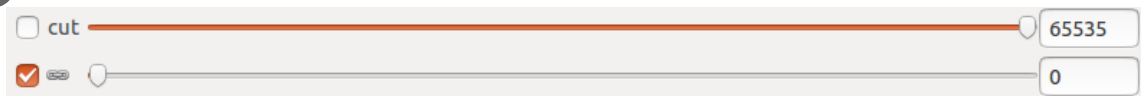


see statistics [p.107]

14 Channel

Displayed channel

15 sliders



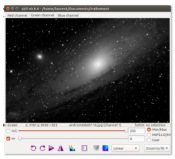
The sliders at the bottom control thresholds for black and white on the grey image, allowing to play with contrast and lightness of each channel separately or together, depending if the chain button is checked or not.

> The sliders are disabled when Histogram or AutoStretch display modes are used.

The different display modes

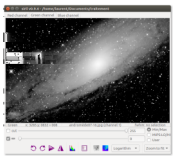
linear

the default mode of Siril. The pixels are displayed from dark to light in a linear scale.



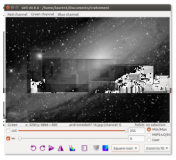
Logarithm

The logarithmic scale. The operation simultaneously exacerbating the weak and light levels of the image.



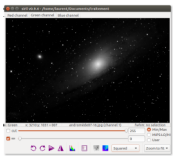
Square root

The square root of each pixel. This can be seen primarily in this mode are the brightest parts of the image.



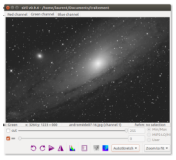
Squared

Square of each pixel. Which can be seen with this model viewer is primarily the most brightest part of the image.



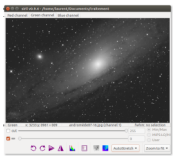
Asinh

The inverse hyperbolic sine is commonly used, it reproduces the perceptive capacity of the human eye, allowing to perceive significantly different brightness levels simultaneously. The asinh function is close to the logarithmic mode but has a better behavior around zero.



AutoStretch

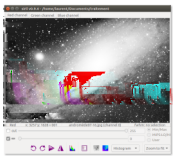
Siril performs auto stretching curves to adjust the image and make it visible on the screen.



Histogramme

The histogram equalization increases the contrast of the image by increasing the dynamic range of the intensity given to the pixels with the most likely intensity values.

It is highly recommended to evaluate all the signals contained in the image.



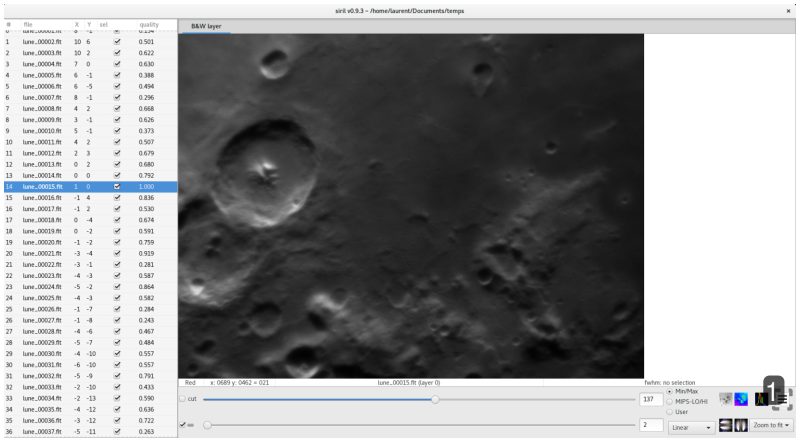
cut

Cut

Instead of keeping pixels with values greater than the value "hi" white when checked: displays black pixels when saturated

All of these modes can also be applied independently on each channel when the bond inter channel is disabled.

1.9.1. Images list



1

Show / hide the image list

List images of the sequence, with their status and quality information.

Sort images by clicking on the title of their column, for example by names, by the FWHM or quality.

Quality is FWHM[p.189] when PSF registration is used, so lower is better

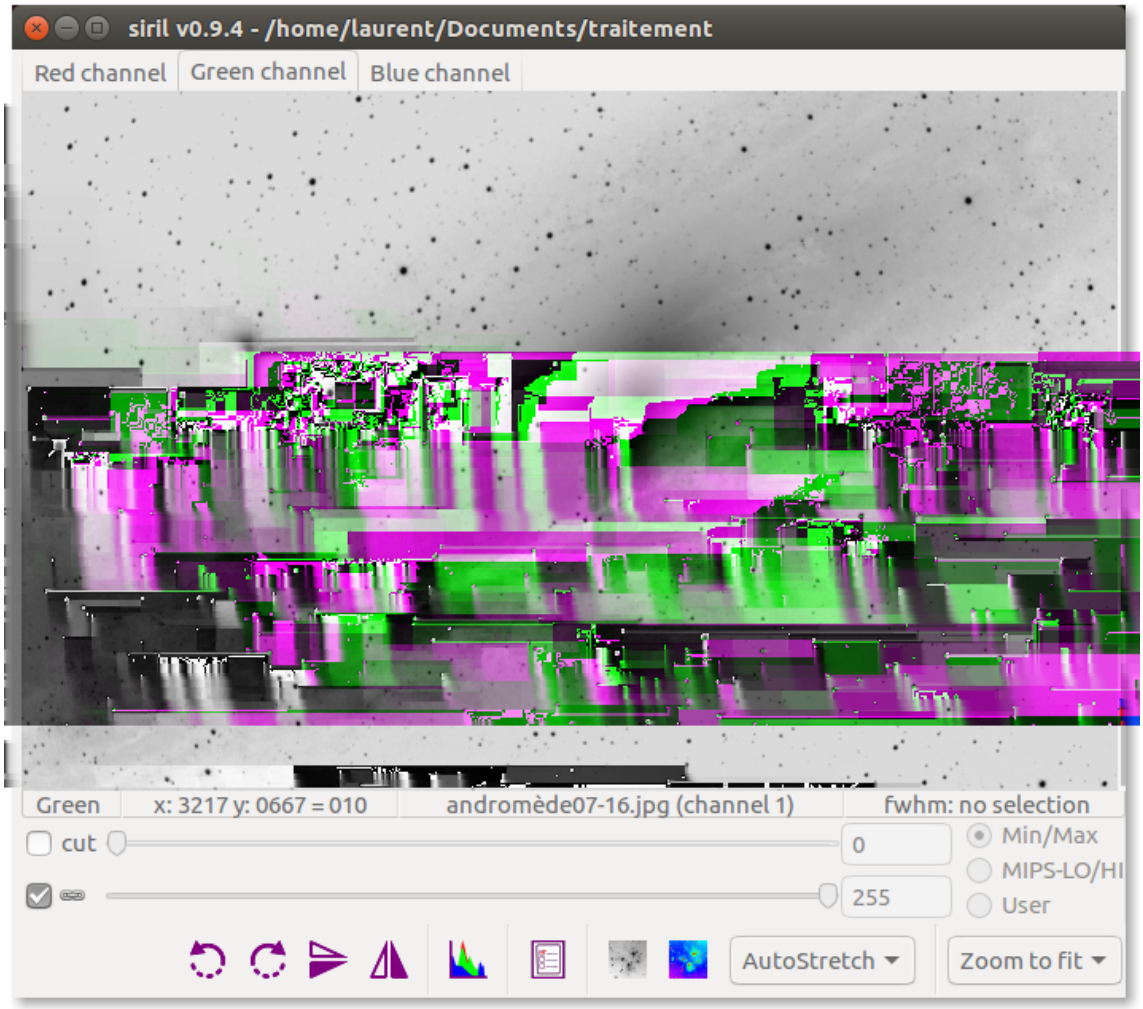
And gradient estimator for DFT[https://en.wikipedia.org/wiki/Cross-correlation] registration, higher is better.

Columns X and Y are the translation in pixels relative to the reference image

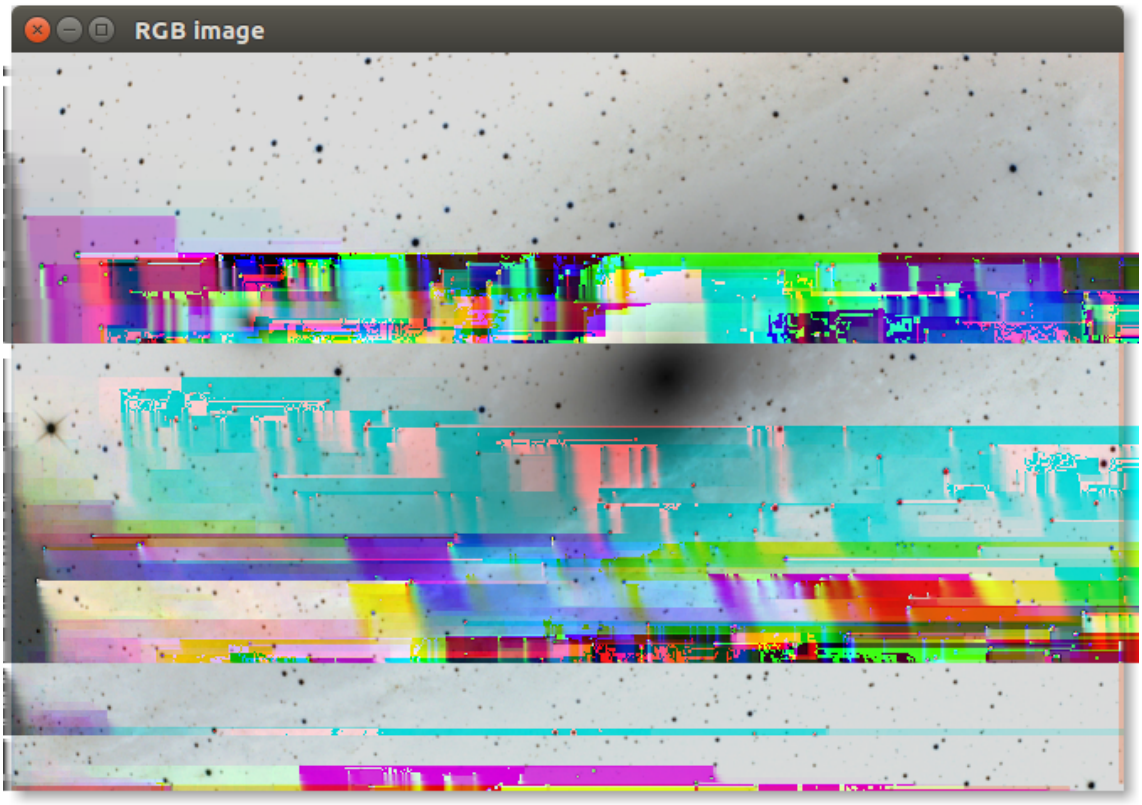
1.9.2. Colors maps viewer

Siril provides different color map viewer, including rendering false color.

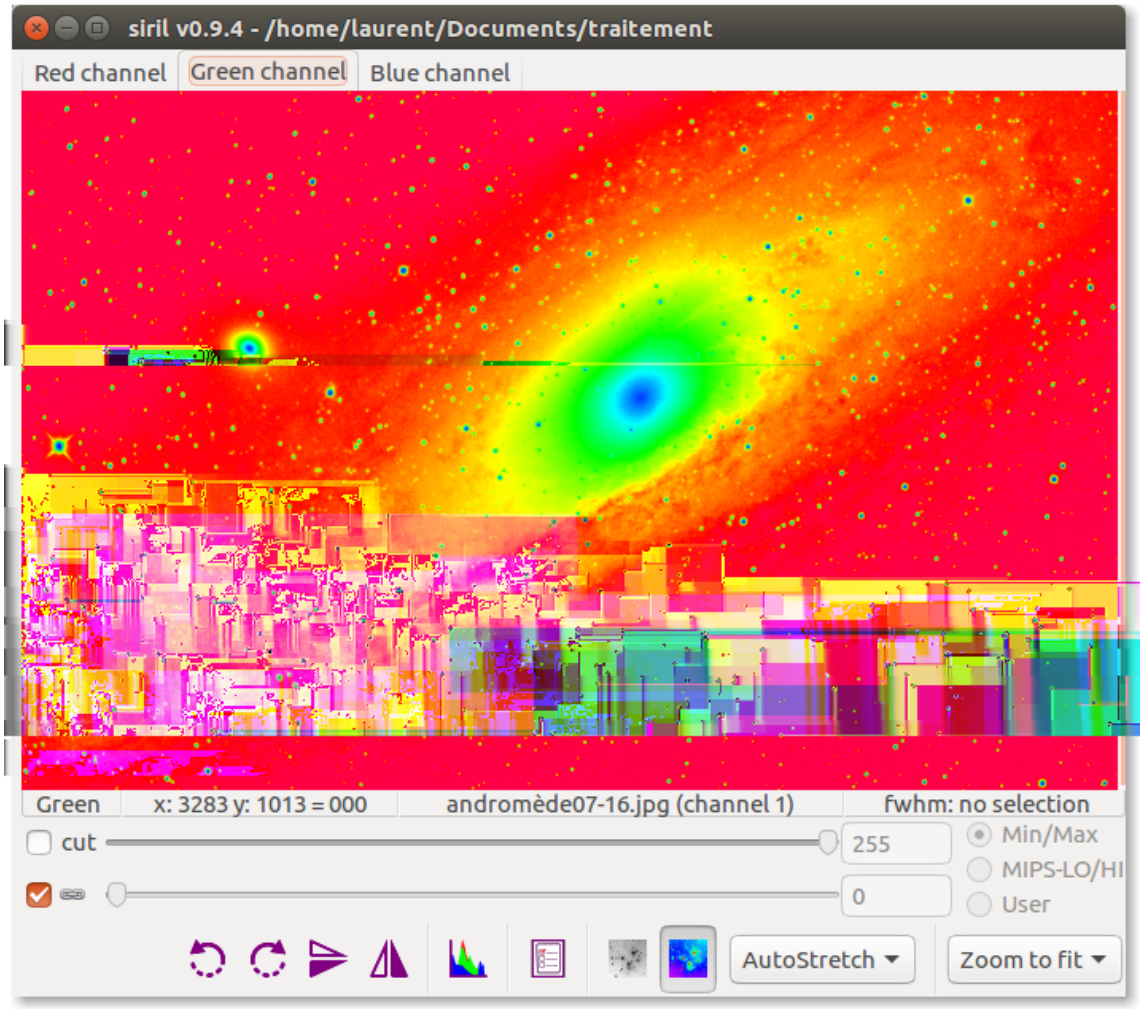
Negative view



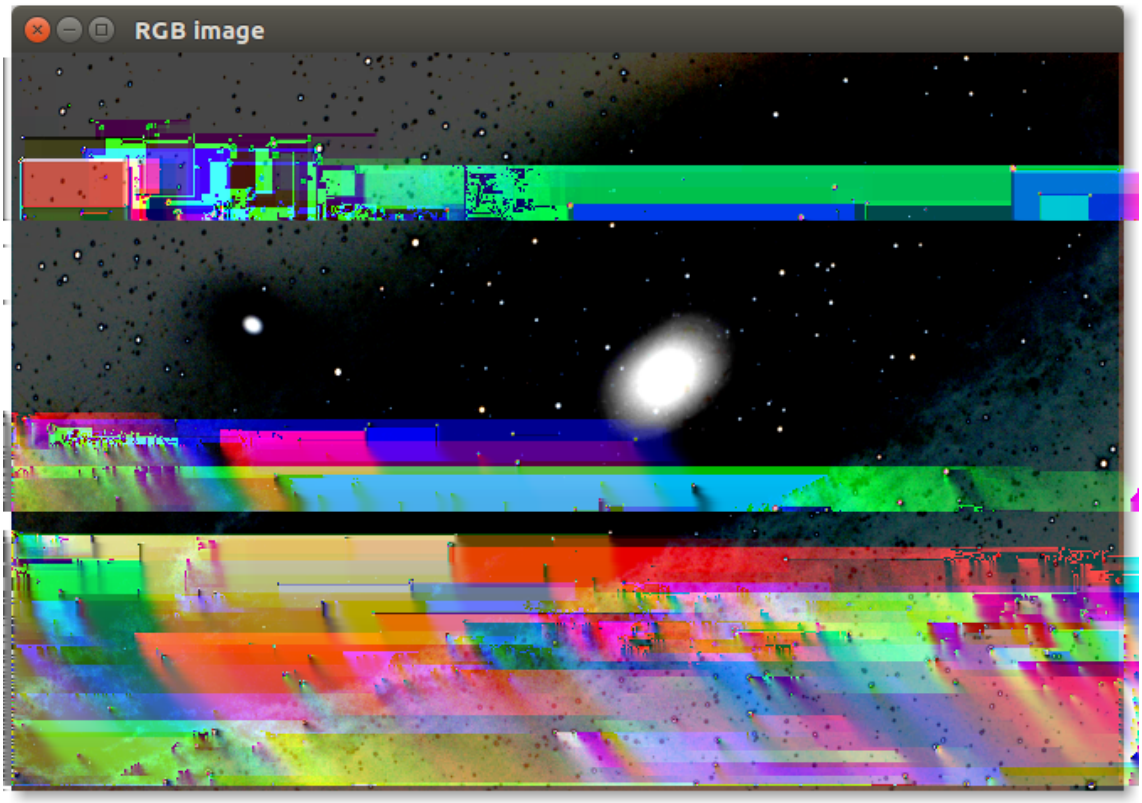
RVB image negative view



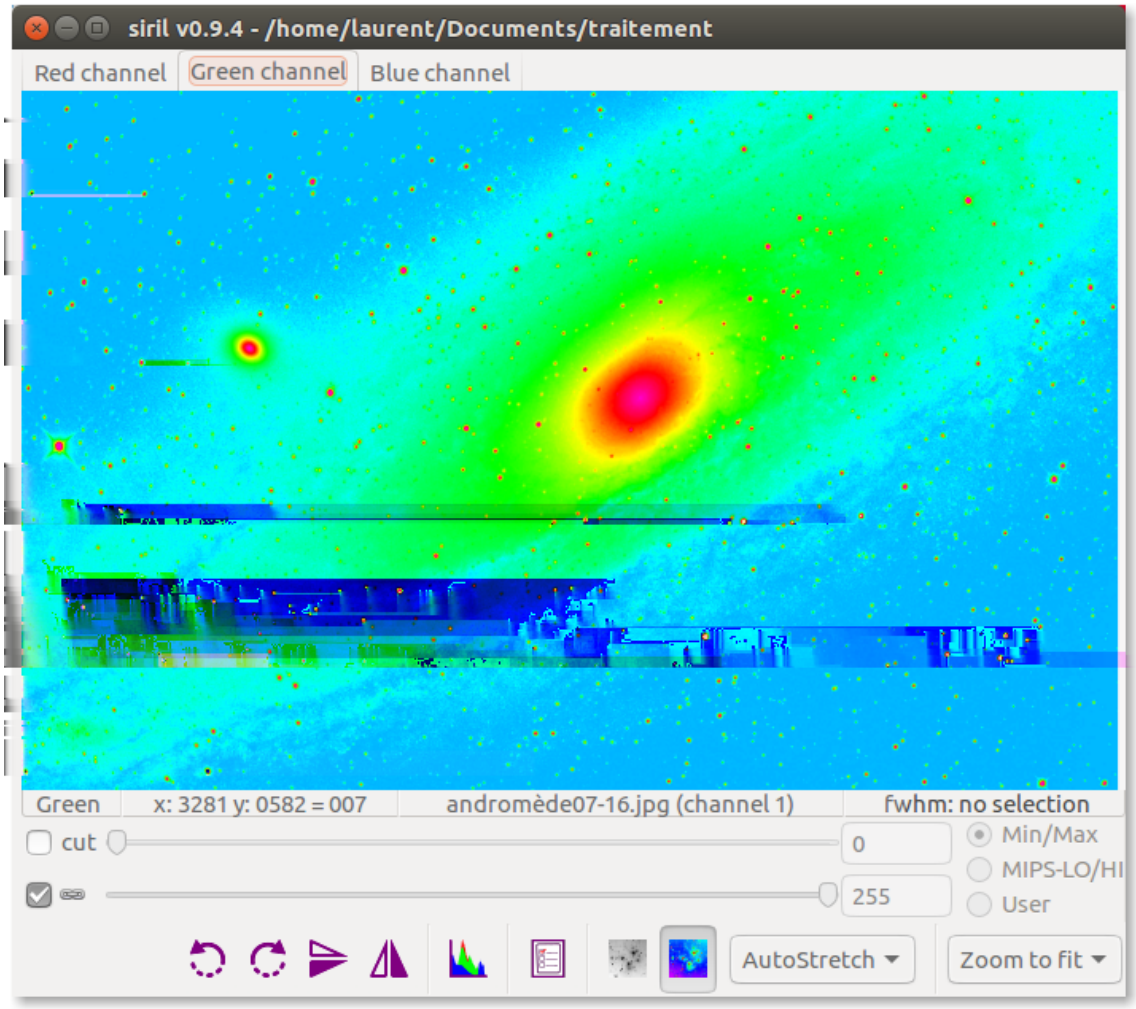
Rainbow color-map (false color rendering)



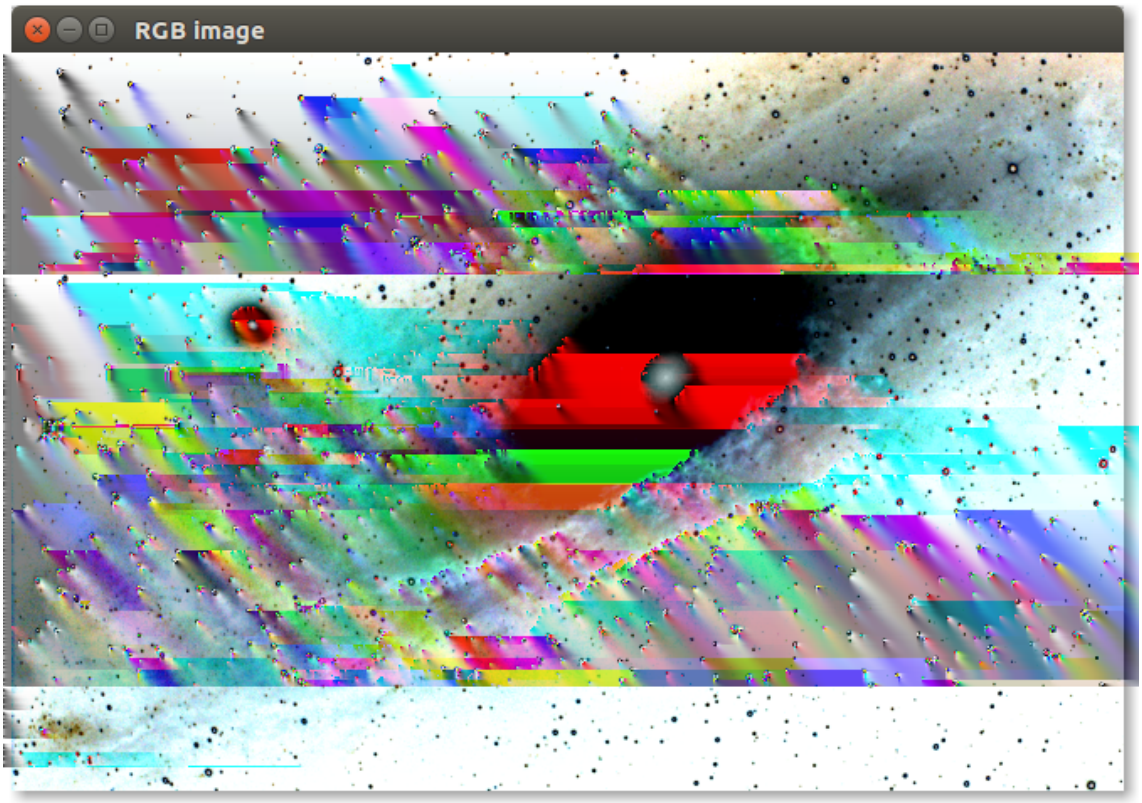
RVB image false color rendering



Negative view + Rainbow color-map



RVB image Negative view + false color rendering



1.9.3. Histograms

You have to stretch the histogram playing mainly on mid-tones and shadows (rarely highlights).

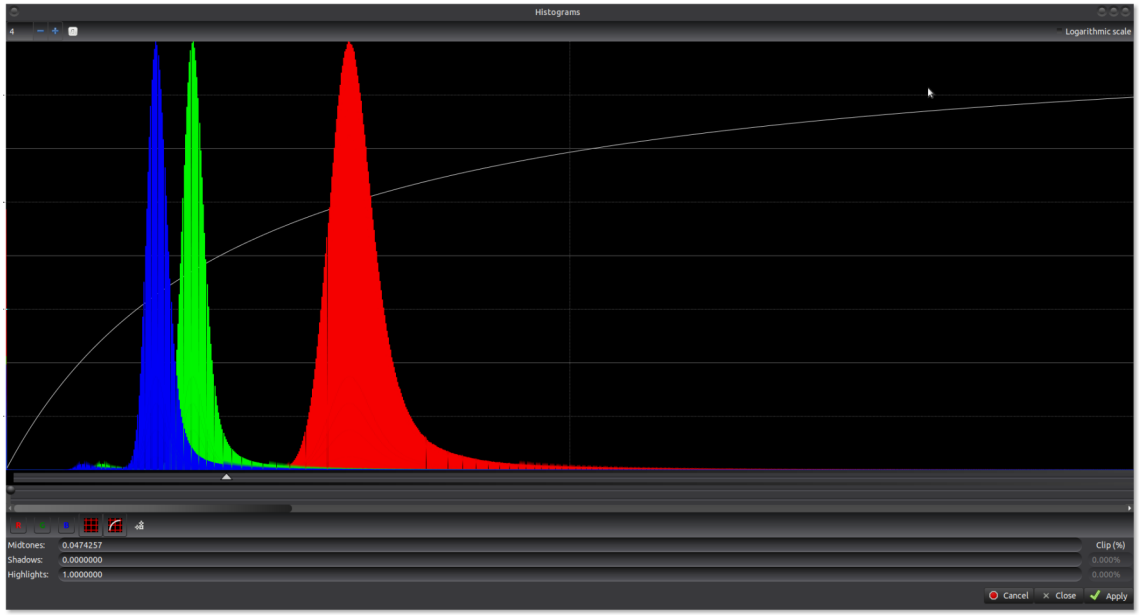
The mid-tones stretch, while the shadows reposition the histogram to the beginning, to not have a too clear sky.

Avoid having too much loss of pixels because it is losing information.

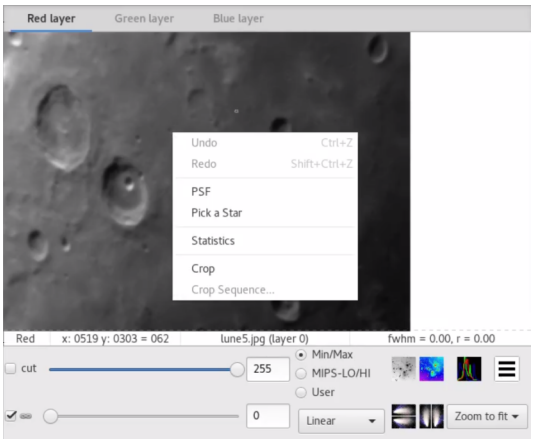
The icon with the toothed wheel allows to transform the histogram in the same way as does the autostetch.

Usually it gives good stuff, but sometimes it clips too many pixels.

Histogram



1.9.4. Right click functions



The following functions are accessible

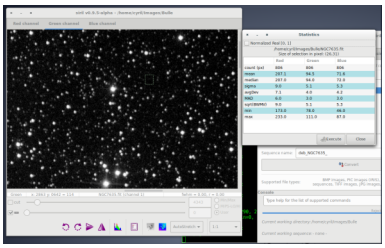
- Undo: Cancel last operation processed. Warning, does not work with command line functions
- Redo: Restore last operation canceled. Warning, does not work with command line functions
- PSF and Pick a Star (see Photometry [p.100])
- Statistics: Show all computed statistics
- Crop: Crop the image from selection
- Crop sequence: crop the image sequence from selection

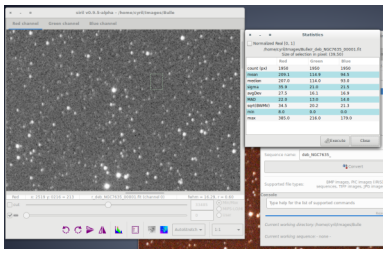
1.9.5. Display use

The images below picture the result in Siril using the **Auto-Stretch** rendering mode.

Note the improvement of the signal-to-noise ratio regarding the result given for one frame in the previous step (take a look to the sigma value).

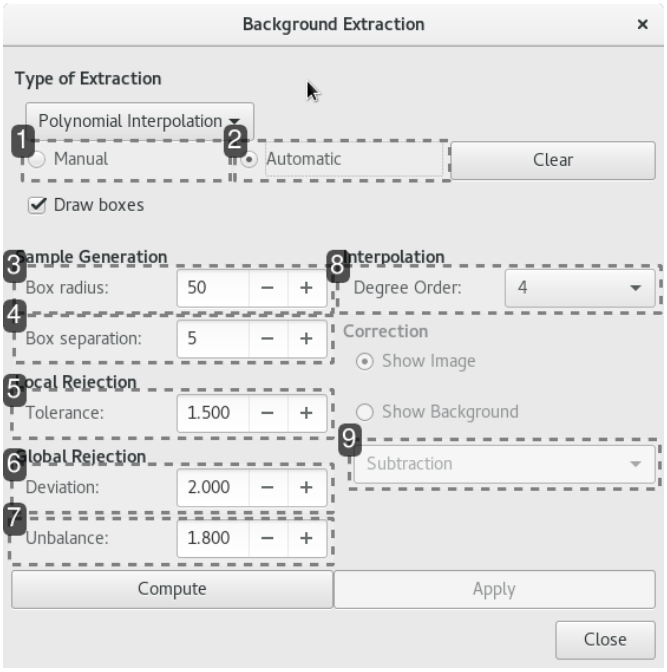
The increase in SNR is of 21/5.1=4.11≈1√12=3.46 and you should try to improve this result adjusting σ_{low} and σ_{high}.



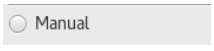


1.10. Image Processing

1.10.1. Background extraction



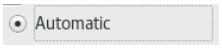
1 Manual



You must click on the image to choose your own background samples.

The only parameters are the sample size (box radius) and degree of order (degree Order)

2 Automatic



Siril take background samples automatically, with the parameters:

- sample Generation
- local Rejection
- Global Rejection

3 Box radius

Radius of the sample area

Pixel Radius of a background sample area. Multiply by 2 to get the length of the zone.

4 Box separation

Distance between zones

Distance in pixels between two adjacent zones of background sample.

5 Tolerance

Parameter that indicates the local tolerance of the sample rejection in sigma units.

6 Deviation

Déviation

Parameter that indicates the overall tolerance of the sample rejection in sigma units.

7 Unbalance

The factor is multiplied by the deviation parameter to include more dark pixels in the background evaluation.

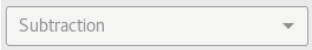
8 Degree Order

Choose the degree of order of the polynomial used in the adjustment.

The default value is 4.

The higher the degree of order, the greater the gradient can be suppressed.

9 Correction Type

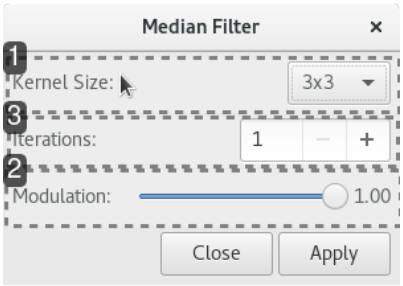


Choose the type of correction you want to apply:

Subtraction: It is mainly used to correct additive effects such as gradients caused by light pollution or the Moon.

Division: It is mainly used to correct multiplicative phenomena such as vignetting or differential atmospheric absorption for example. However, this type of operation must be performed for the correction of 👉master-flat [p.122]

1.10.2. Median filter



1 Kernel Size

Odd pixel size of the neighborhood of each pixel.

possible sizes

- 3x3
- 5x5
- 7x7
- 9x9
- 11x11
- 13x13
- 15x15

2 Modulation

Modulation parameter defined in the range [0.1].

When this setting is 1, no modulation is applied.

If the parameter is null no filter is applied.

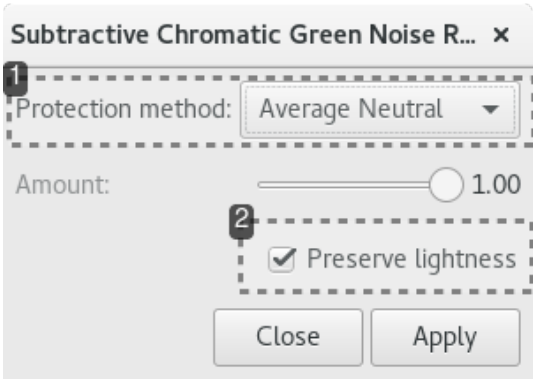
3 Iterations

Number of iterations of the algorithm.

This is useful when the modulation parameter is different from 1 to help the whole process to converge to a better result.

1.10.3. Reducing green tint in processed images

Subtractive Chromatic Green Noise Reduction



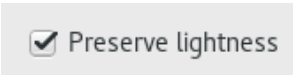
1 Protection method

Average Neutral Protection is the default option. This is suitable in most cases.

Possible values

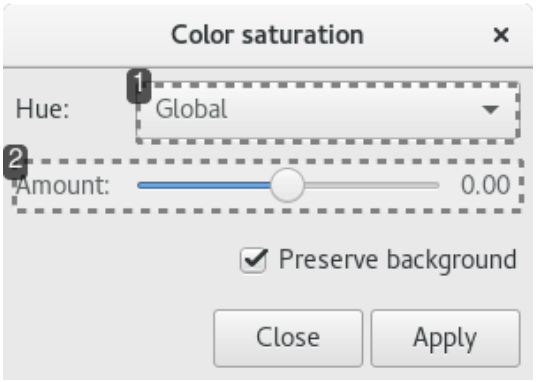
- Average Neutral
- Maximum Neutral
- Maximum Mask
- Additive Mask

2 Luminance



Preserves [p.186] original CIE L*[p.186] component of the processed image.

1.10.4. Color saturation



1 Color

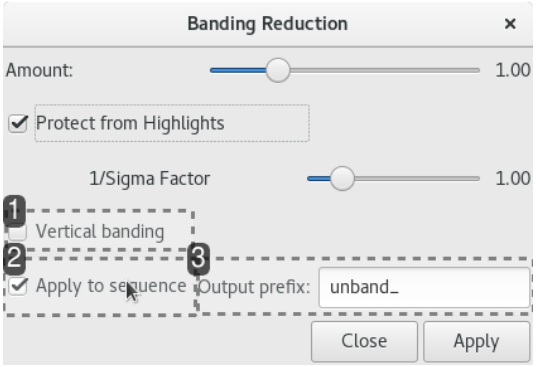
Pink-Red / Red-Orange
Orange-Brown / Yellow
Yellow-Green / Green-Cyan
Cyan-Blue / Blue-Magenta
Magenta / Pink
Global

2 Amount

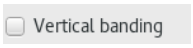
A value of 0 means no change, the negative values reduce color saturation, positive values increase them.

1.10.5. Banding Reduction

Banding[p.185] Reduction

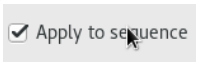


1



Check this box if the banding[p.185] is vertical.

2



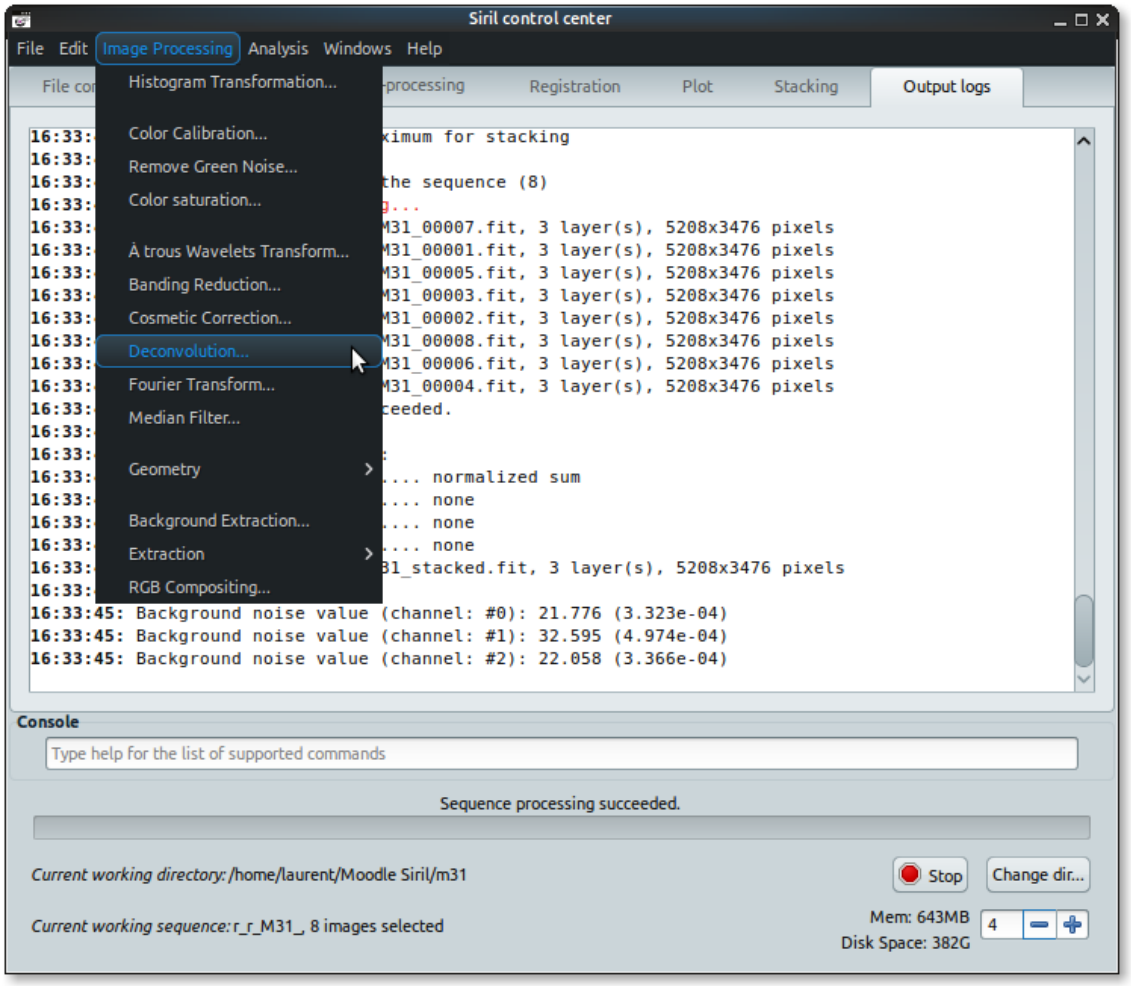
If a sequence is loaded and this function is activated, performs processing on all images of the sequence instead of the single image loaded.

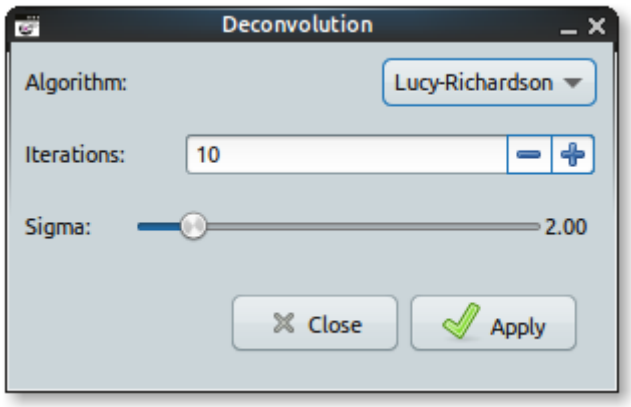
3

Prefix of unband images names

> The canon banding removal tool can be also used on vertical banding noise.

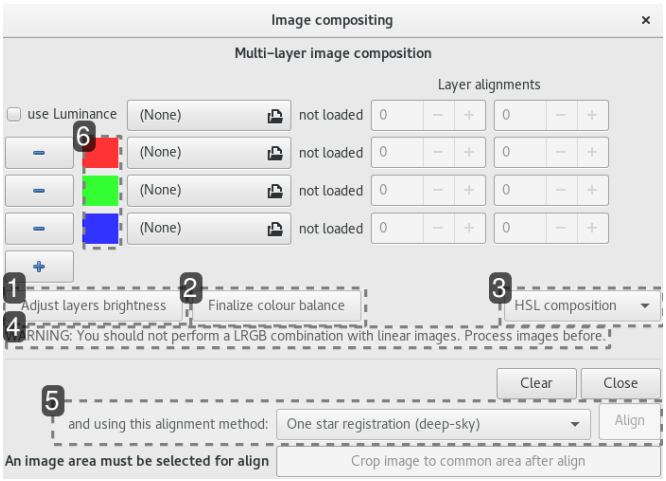
1.10.6. Déconvolution





Since version 0.9.8 a deconvolution algorithm[*https://en.wikipedia.org/wiki/Richardson%E2%80%93Lucy_deconvolution*] has been added

1.10.7. RGB compositing...



1 Brightness Adjustment Layers

Automatically adjust the brightness of the colors chosen for the layers so that they do not saturate the compositing (based on the max value for each image).

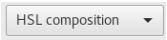
The color shades are unchanged.

2 Finalize the color balance

Opens the RGB calibration tool on the resulting image. This will alter the resulting image in a destructive way to align the three channels thereof.

Changing the settings in this window after cancels the color balance: it must be redone.

3 Type of color composition



Type of color composition based on luminance. They are the way to add color information for the luminance data.

Possible values

HSL[p.190] composition

HSV[p.191] composition

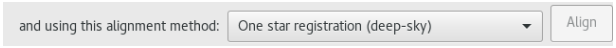
CIELAB[p.186] composition

4 Warning

⚠

You should not make a LRVB combination with linear images. Treat first images.

5 Automatic alignment



Automatic alignment using the selected method.

Possible values

Image pattern alignment (planetary/deep-sky)

One star registration (deep-sky)

⚠

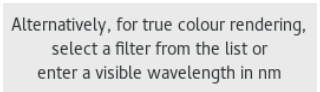
For LRGB composition, the reference layer must be luminance.

6 Color selection

Dialog box Select a Color

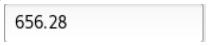
Select a Color

1



For true color rendering, select a filter from the list or enter a visible wavelength in nanometers.

2



Enter a wavelength value in nanometers here, between 380 and 780, the decimal values are accepted.

The LRVB alignment does not take into account rotation.

To work around the problem create a sequence with LRVB image_01, Image_02, etc ....

Then you align the sequence globally. New aligned files will be created that you can easily integrate into the LRVB alignment

1.10.8. Extraction

1.10.8.1. Channel extraction

The Channel extraction allows you to obtain individual RGB pixel channels as grayscale FIT images.

Channel extraction does not modify the image to which it is applied.

It can only be applied to color images.

Supported color spaces

RGB[p.194]

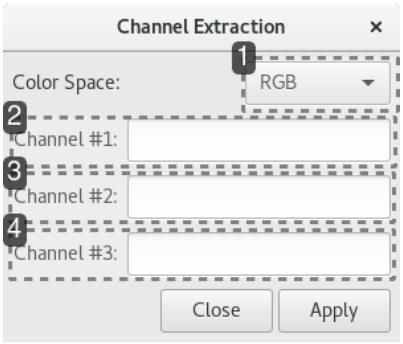
HSL[p.190]

HSV[p.191]

CIE L*A*B*[p.191]

Select a color space to extract one or more channels or components.

Default, is the RGB color space.



1 Color space

Select the color space for the three images from

Possible values:

RGB[p.194]

HSL[p.190]

HSV[p.191]

CIE L*a*b*[p.186]

2 Channel #1

Name the first FITS file for backup.

3 Channel #2

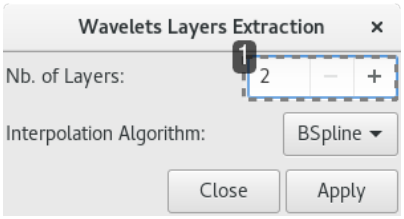
Name of the second FITS file for backup.

4 Channel #2

Name of third FITS file for backup.

1.10.8.2. Wavelets

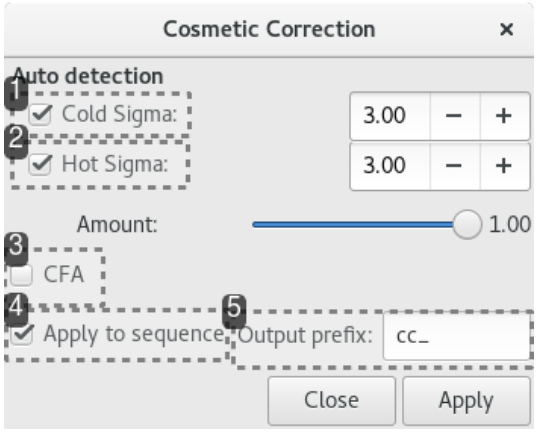
Wavelets extraction layers



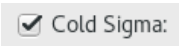
1

Number of Layers

1.10.9. Cosmetic correction



1



How many times (in average deviation units) the pixel value must be different from the value of its surrounding neighbors to be considered a cold pixel.

2

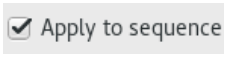
How often (in mean deviation units) the pixel value must be different from the value of its surrounding neighbors to be considered a hot pixel.

3

Check if the target layers are CFA images[p.186].

Uncheck if the images are from a monochrome sensor.

4



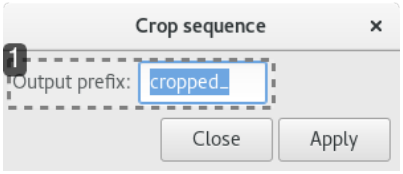
If a sequence is loaded and checked, performs processing on all images of the sequence instead of the single image loaded.

5

Prefix image names resulting from aesthetic correction process.

1.10.10. Sequence crop

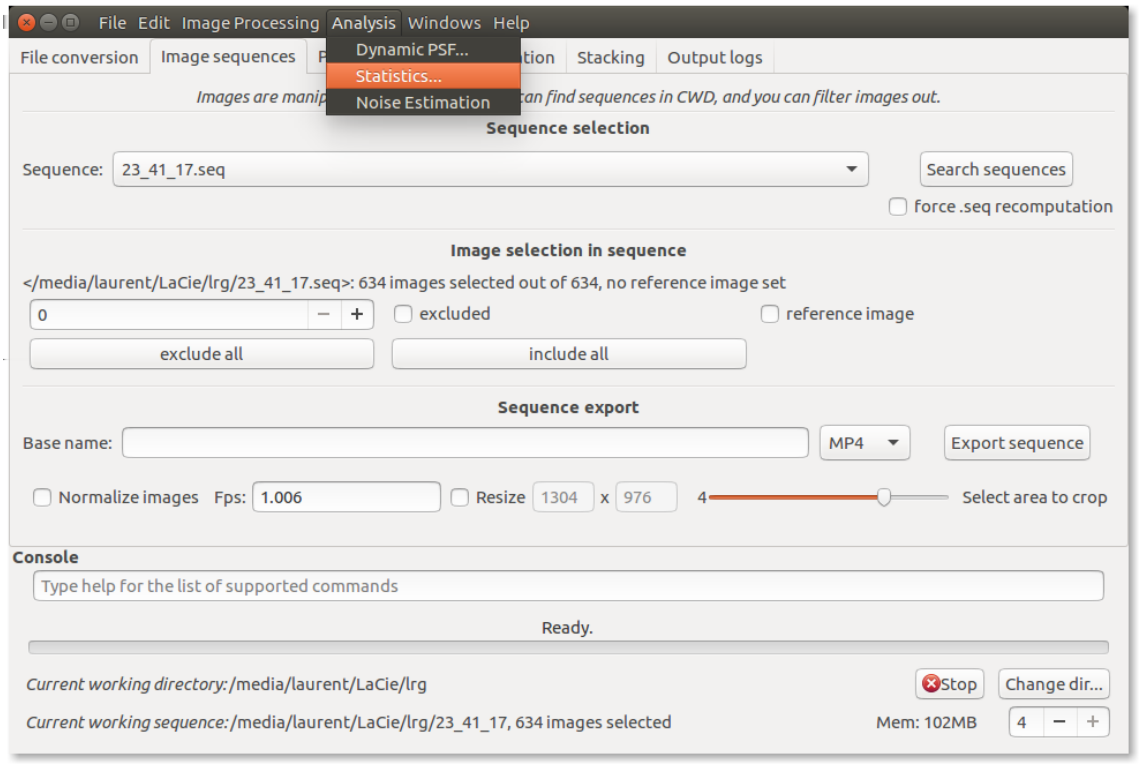
Crop sequence



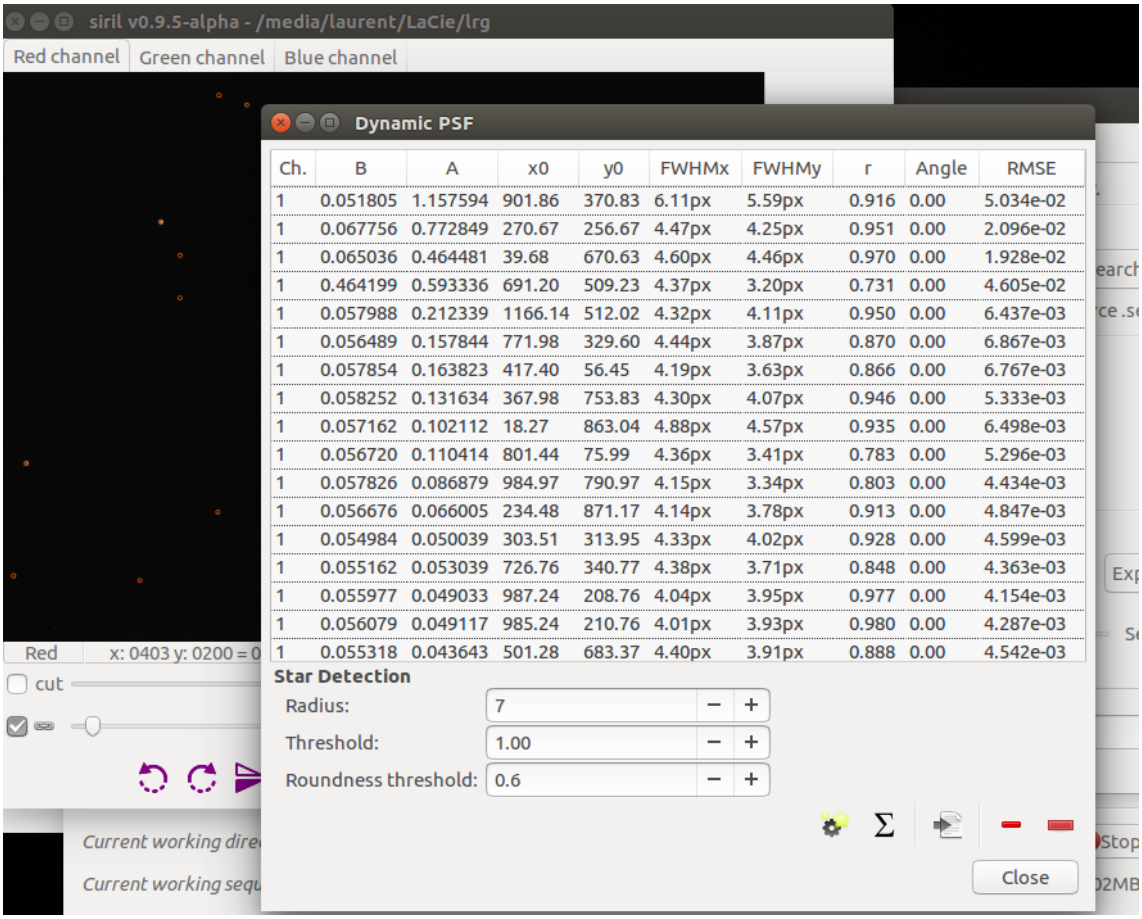
1

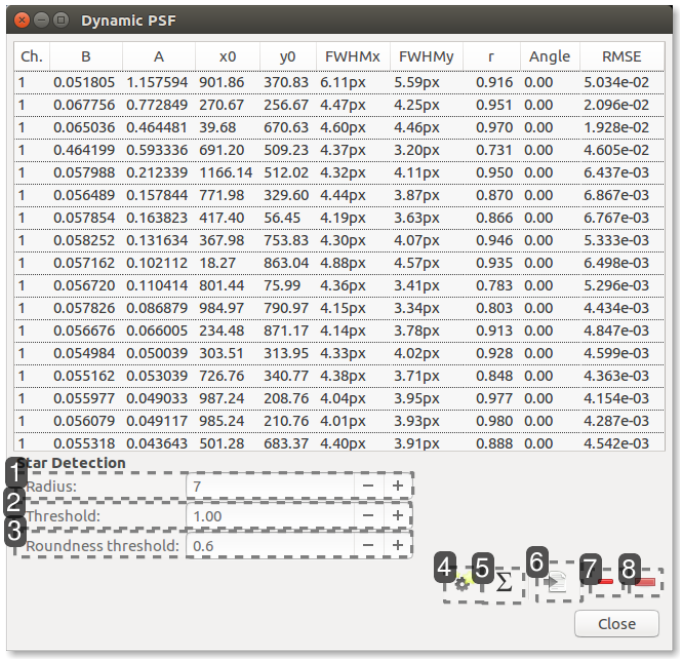
prefix of the resulting sequence of the cropping process.

1.11. Analysis

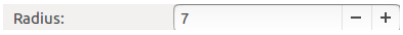


1.11.1. Point Spread Function



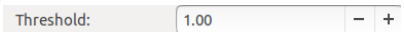


1 Radius



Size in pixels of the search box.

2 Threshold value

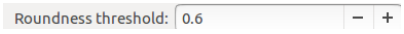


Threshold value above the noise for stars detection.

The median and standard deviation values of the image are determined, then the threshold is: `median + [threshold] x stdev`.

> The default value for threshold is 1.00.

3 Threshold value



Threshold value of star roundnesses. Decrease this value if your stars show small trails.

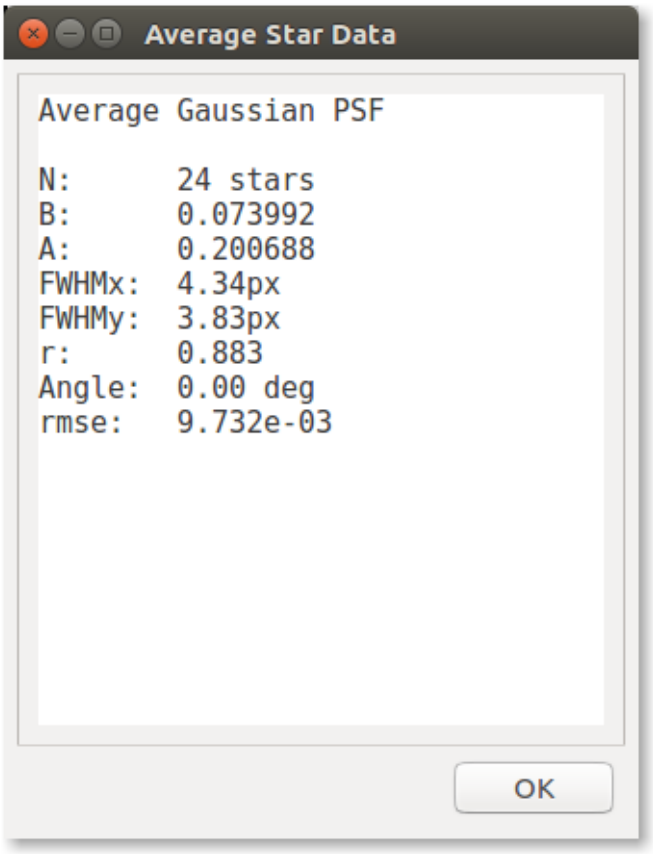
> Ensure you have a good reason to change this value because it may degrade the star detection.

4 Star Detection

Detect stars in the current loaded image (not sequence image) and compute PSF

5 PSF parameters

Average PSF parameters for selected stars



6 Export PSF

Export PSF parameters in a ASCII file

★

Save as *stars.lst* in the current working directory

7 Remove

Remove a star from list

8 Remove all

Remove all stars from list

The PSF function uses the Levenberg-Marquardt[*https://en.wikipedia.org/wiki/Levenberg%E2%80%93Marquardt_algorithm*] for minimization.

Dynamic PSF

Dynamic PSF is a dynamic tool inspired by the PixInsight routine of the same name. It is used to fit unsaturated stars within the image.

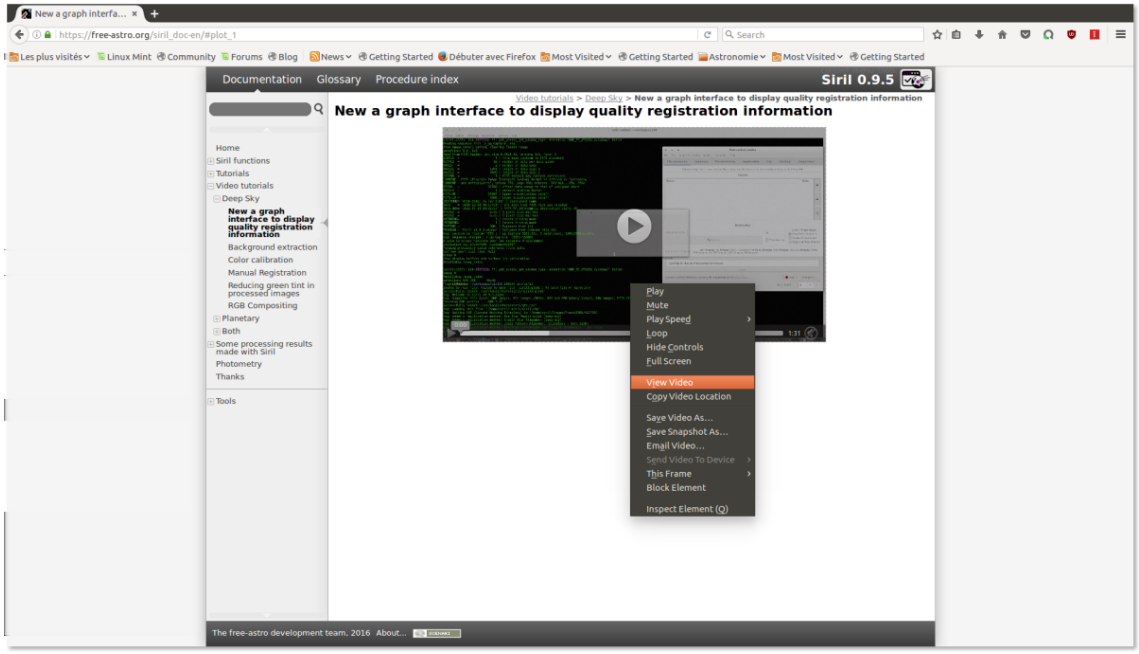
⚠

For now, the PSF model is restricted to a 2D Gaussian function which is sufficient in most cases.

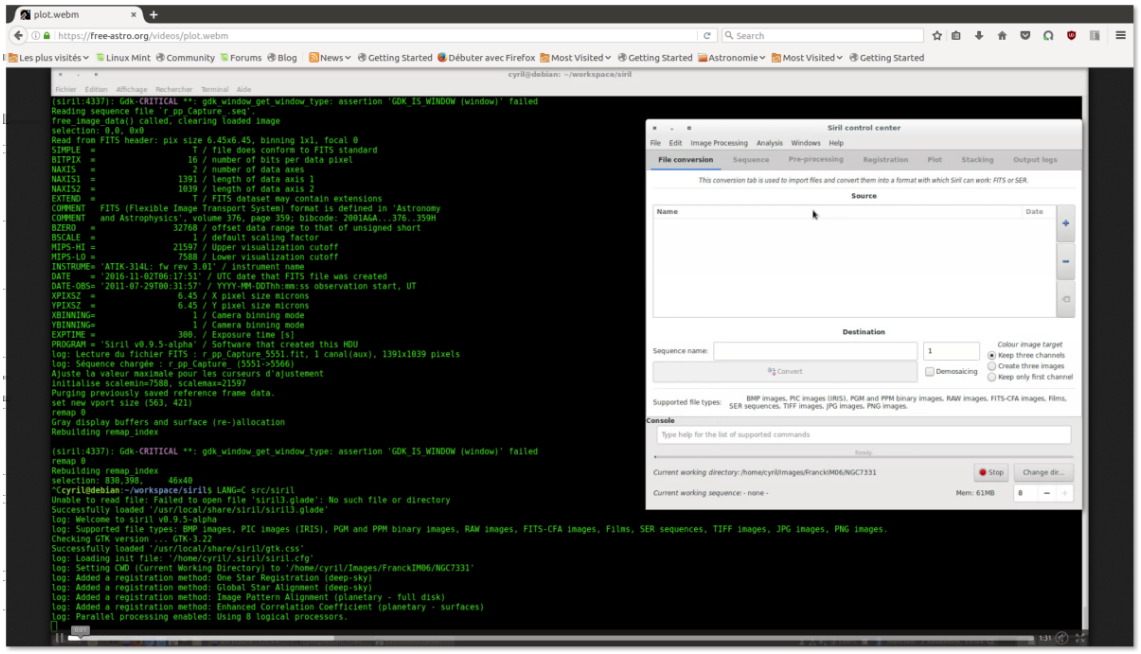
Dynamic PSF

View full-screen video

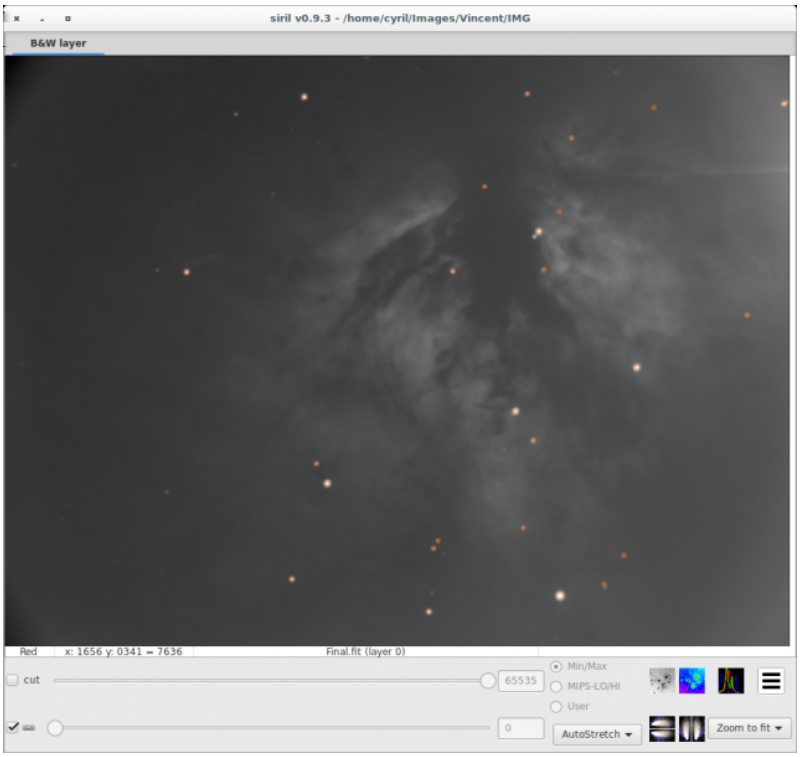
Right click and choose "view video"



In your browser, click go back one page to exit



Example star detection in an image



The stars are closely surrounded by orange circles

2D Gaussian

$$G(x,y)=B+Ae^{-\left(\frac{(x-x_0)^2}{2\sigma_x^2}+\frac{(y-y_0)^2}{2\sigma_y^2}\right)}$$

Parameters:

- B : The average local background in the [0, 1] range.
- A : The maximal intensity of the star in the [0, 1] range: this is the peak value of the fitted function, located at the centroid coordinates x0 and y0.
- x0 and y0 : The centroid coordinates in pixel units, which is the position of the center of symmetry of the fitted PSF.
- $FWHM_X$ and $FWHM_Y$: The Full Width Half Maximum on the X and Y axis in pixel units.

  These parameters are calculated as follow :
  - $FWHM_X = 2\sigma x\sqrt{2\log 2}$
  - $FWHM_Y = 2\sigma y\sqrt{2\log 2}$
- It is possible to obtain the FWHM parameters in arcseconds units. This requires you fill all fields corresponding to your camera and lens/telescope focal in the setting parameter window. If standard FITS keywords FOCALLEN, XPIXSZ, YPIXSZ, XBINNING and YBINNING are read in the FITS HDU, the PSF will also compute the image scale in arcseconds per pixel.
- r: The roundness parameter. It is expressed as $FWHM_Y/FWHM_X$, with $FWHM_X>FWHM_Y$ the symmetry condition.
- Angle: The rotation angle of the X axis with respect to the centroid coordinates in the [-90, 90] range. The angle θ

  is computed as follow:
  - $x' = +x\cos\theta + y\sin\theta$
  - $y' = -x\sin\theta + y\cos\theta$
- RMSE: This is an estimation of fitting quality. The smaller the RMSE is, the better the function is fitted.

Minimization

Minimization is performed with a non-linear Levenberg-Marquardt algorithm*[https://en.wikipedia.org/wiki/Levenberg-Marquardt_algorithm]* thanks to the very robust GNU Scientific Library*[http://www.csse.uwa.edu.au/programming/gsl-1.0/gsl-ref_35.html]*.

As a first step, the algorithm runs a set of parameters excluding rotation angle in order to set good start values and thus, avoiding possible divergence. If $\sigma x - \sigma y > 0.01$ (parameters directly computed in the 2-D Gaussian formula, see above), then another fit is run with the angle parameter.

Therefore, the Siril Dynamic PSF provides accurate values for all the fitted parameters.

Use

Dynamic PSF can be called from two different ways depending on what you want:

You may want to fit just one star

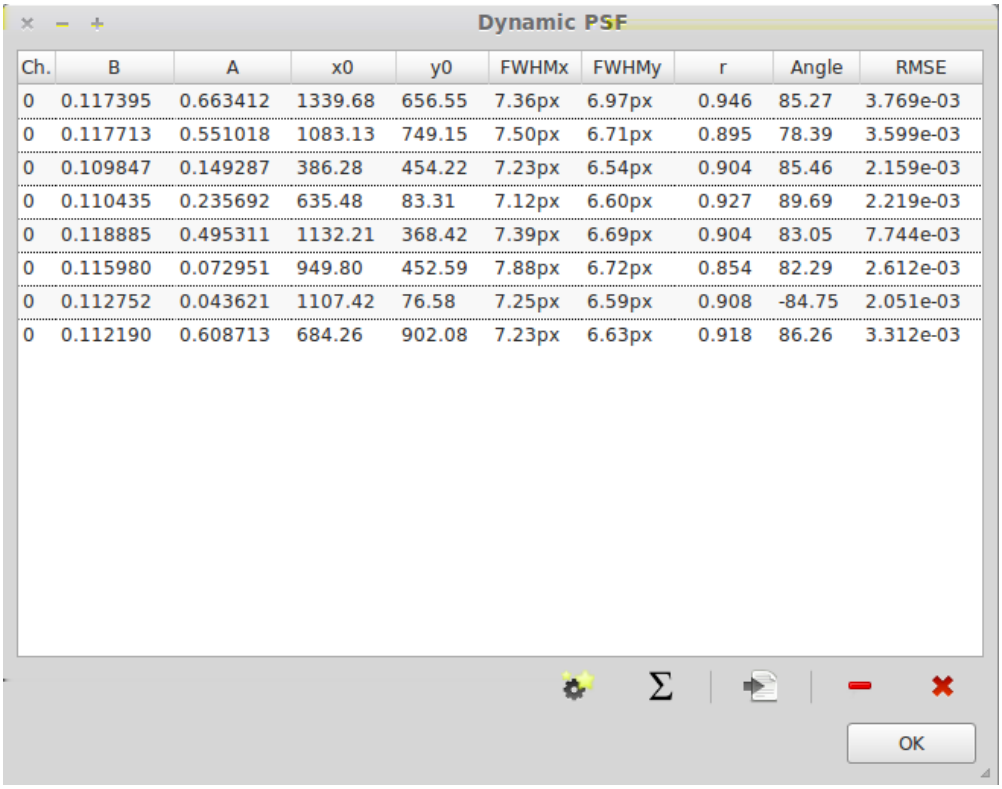
In this case, after drawing a selection around an unsaturated star (this is important for the accuracy of

the result) you can either type "PSF" in the prompt of the console or do a right click and choose the "PSF" item.

In both case you will be able to copy the result and paste it where you want. In this cased a relative magnitude is computed following $-2.5 \times \log I$ where I is the total intensity of the fitted star.

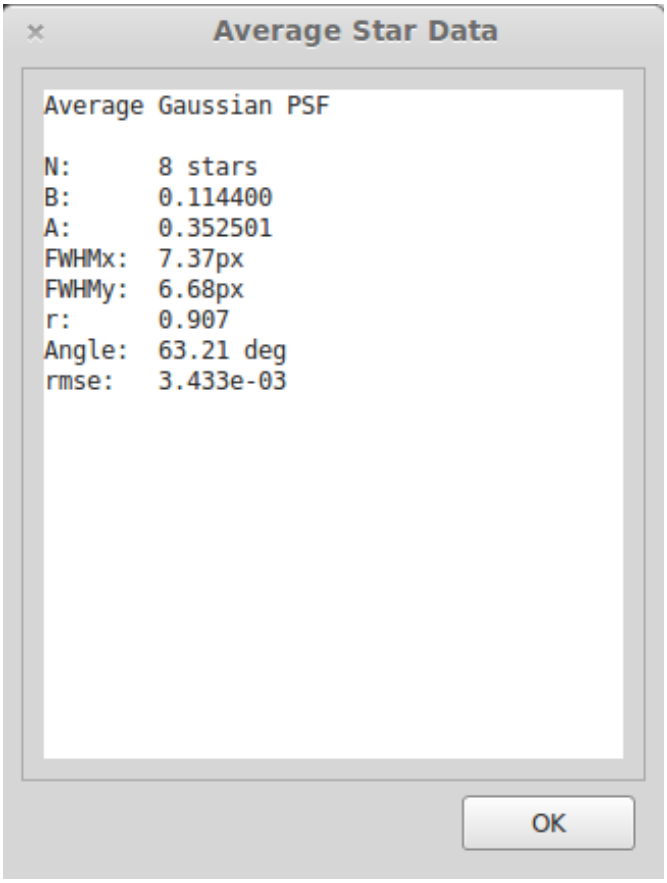
In the case you want to deal with many stars and average it

You **must** work on the dynamic PSF window by ever clicking on the "Analysis" menu and the "Dynamic PSF" item or drawing a selection around an unsaturated star, doing a right click and selecting the "Pick a star" item.



The dynamic PSF window is shown and provides very valuable tools:

- You can select and/or remove a star choosing it in a tree list.
- You can export the result of the PSF in ASCII file.
- You can run the star detection algorithm by clicking on the corresponding button.
- You can average the PSF results and copy and paste the result.



Star detection

As said before, this dynamic PSF also contains an automatic tool of star detection.

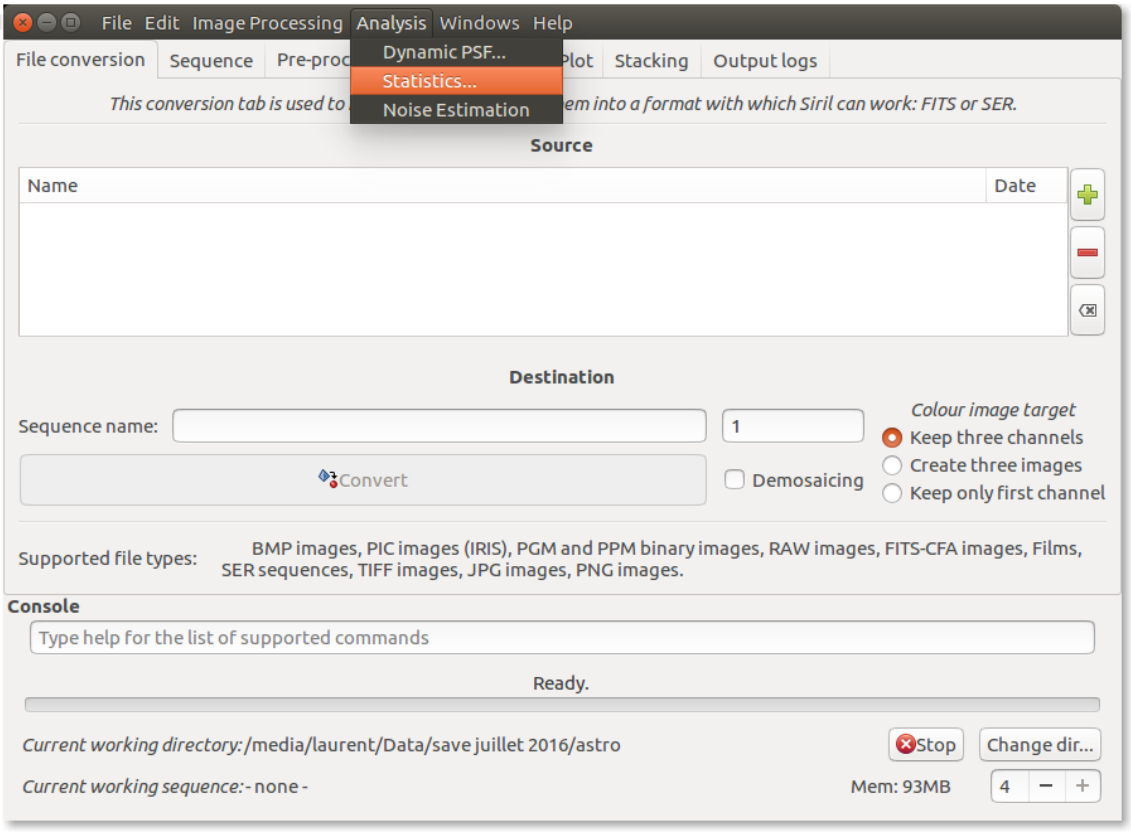
- This algorithm is based on the publication of Mighell, J.J [p.200]. ,which is a simple peak detector algorithm identifying any pixel that is greater than any of its eight neighbours.
- In addition to this basic algorithm, a PSF is applied on every possible "detected stars" and Siril rejects all detected structures that don't fulfil a set of prescribed detection criteria.
- Finally, a circle is drawn around detected stars. Nevertheless, if a non-stellar object is selected, it is possible to select it and remove it from the list.

You may notice that during the automatic star detection, no rotation angle are fitted due to speed consideration.

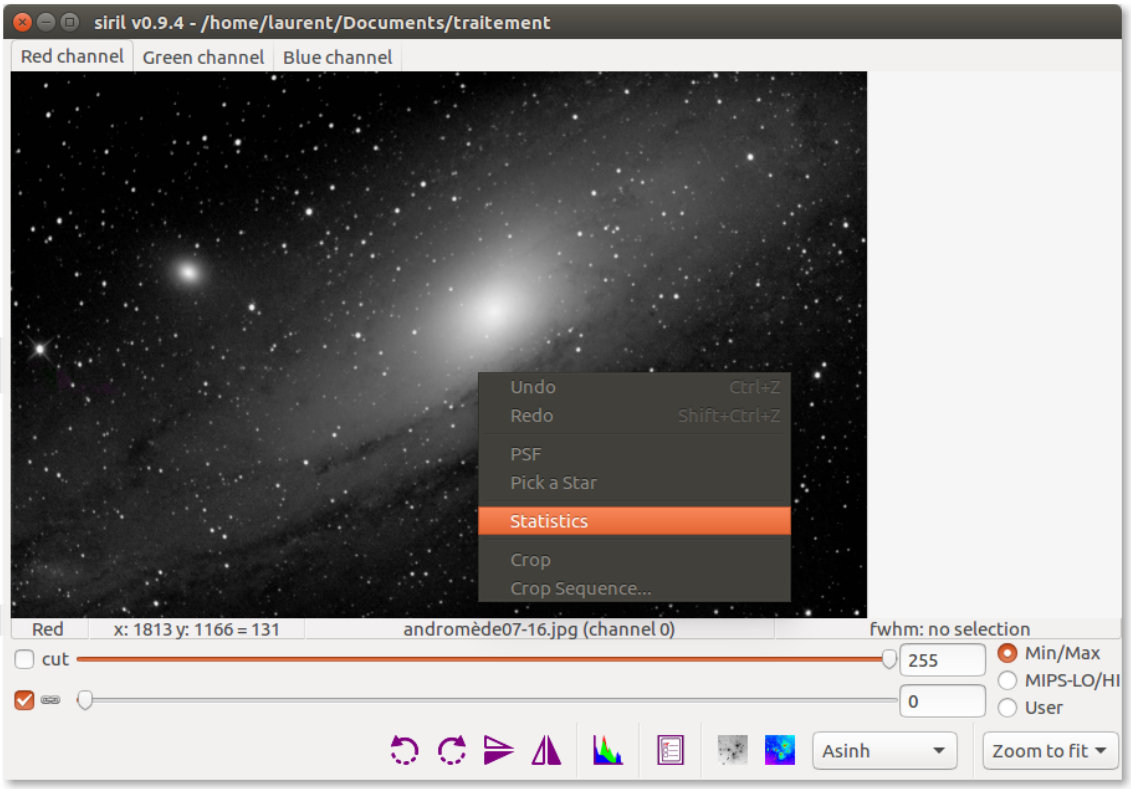
1.11.2. Statistics

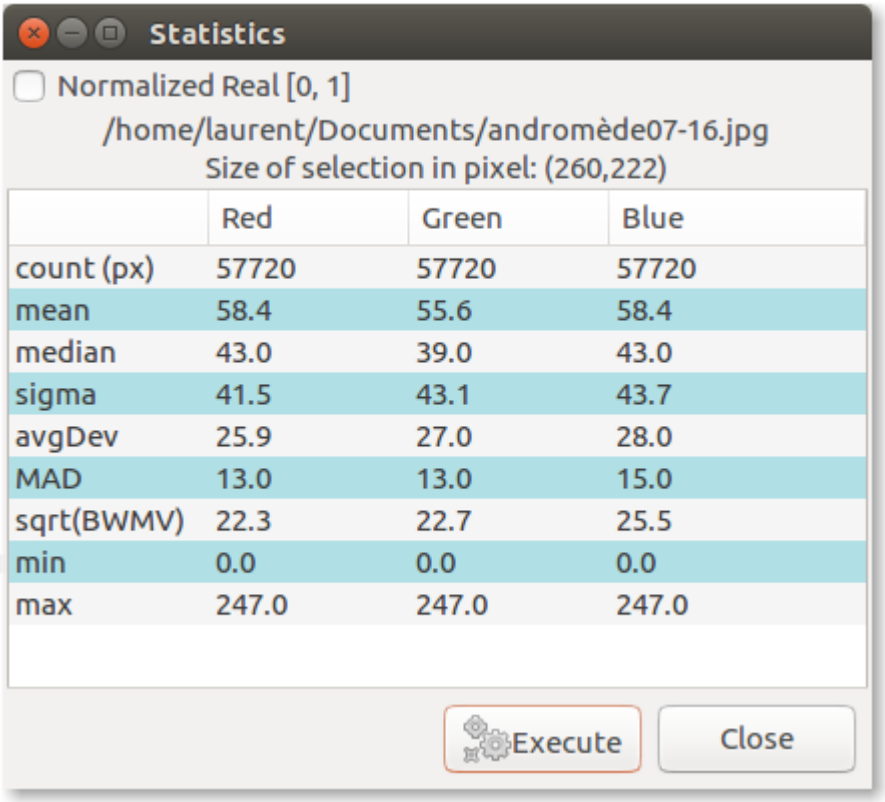
Statistics access

By the menu ☰ Analysis/Statistics...



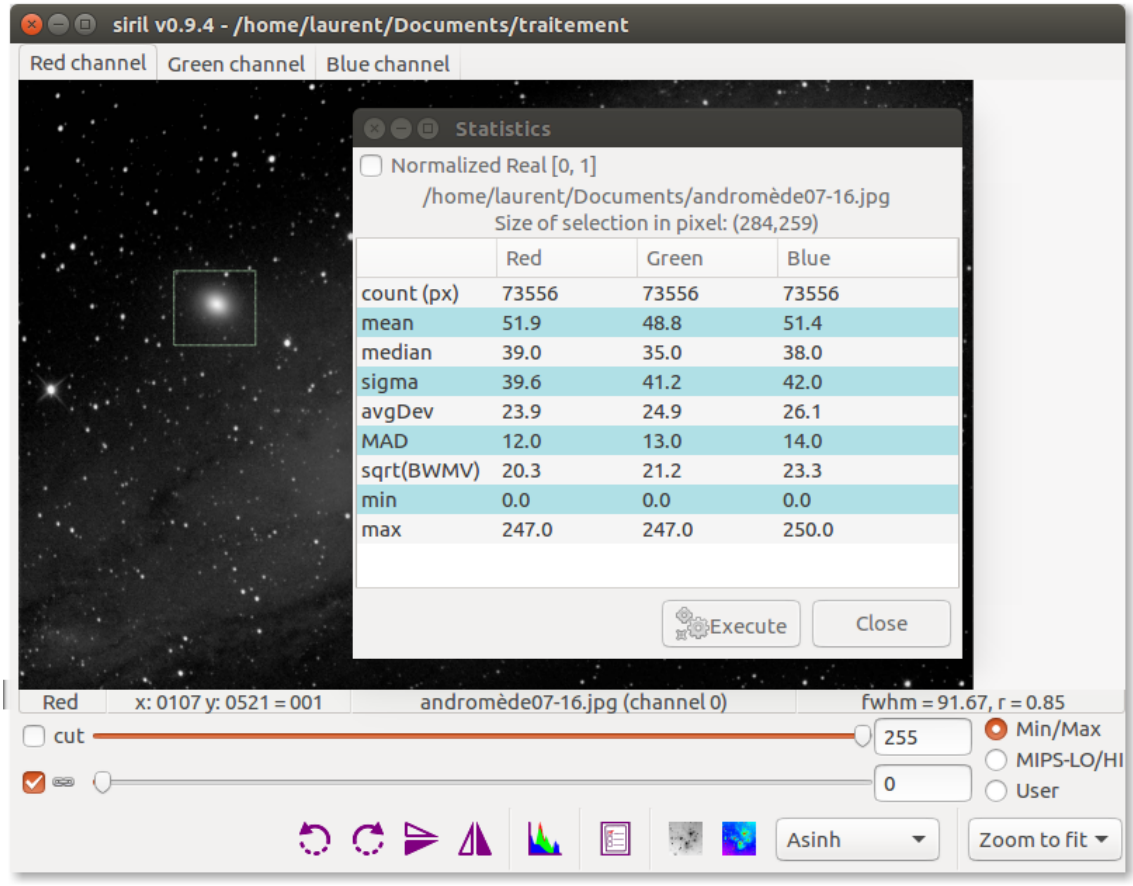
By right click in picture



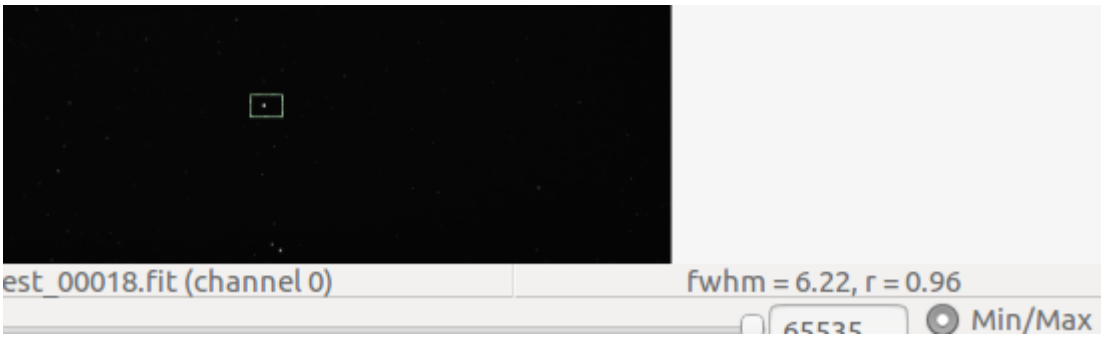


statistics of an image area

If you want to know the statistics of an image area, select it, then click Execute



Selected area



when an area is selected, FWHM and roundness in pixels are displayed

r is the ratio of the FWHM in x and y.

If the value is 1, it means that we have a perfectly round star.

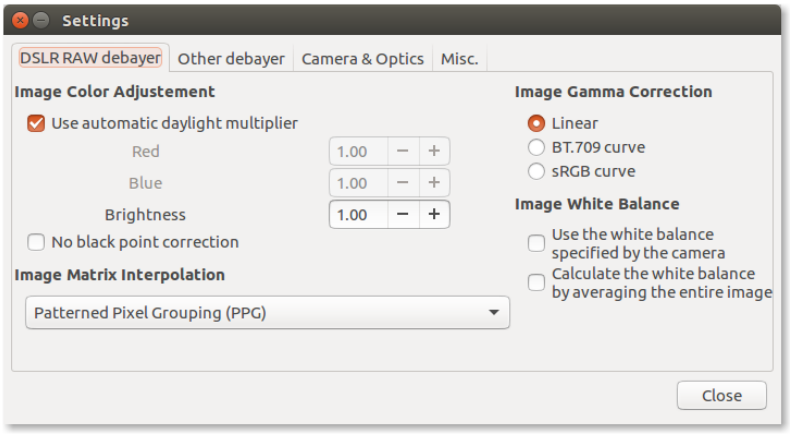
1.11.3. Noise Estimation

Display noise estimation in Output Tab

```
11:44:39: Noise standard deviation: calculating...
11:44:39: Background noise value (channel: #0): 0.548 (2.150e-03)
11:44:39: Background noise value (channel: #1): 0.777 (3.048e-03)
11:44:39: Background noise value (channel: #2): 0.518 (2.033e-03)
11:44:39: Execution time: 100.71 ms.
```

1.12. Settings

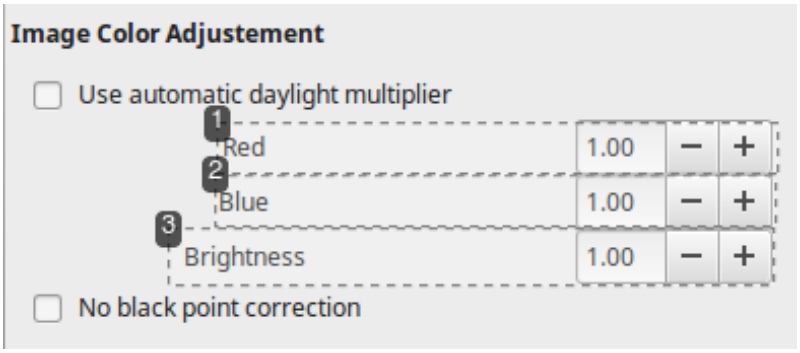
1.12.1. DSLR RAW debayer



1.12.1.1. Image Color Ajustement

☑Use automatic daylight multiplier

When this option is enabled, coefficients for each channel are automatically computed. You can choose your own multiplier by unchecking this box.



1

Red

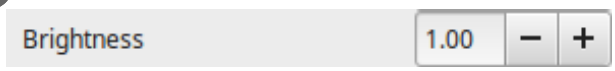
Specify your own raw white balance. The value for the Red scale is relative to the Green.

2

Blue

Specify your own raw white balance. The value for the Blue scale is relative to the Green.

3



Brightness

Divide the white level by this number.

☑No black point correction

When this option is enabled, no black point correction is applied to raw images.

1.12.1.2. Image Matrix Interpolation

Select an interpolation method to perform demosaicing before opening image.

High-speed, low-quality bilinear interpolation method

Bilinear interpolation is quite fast , color error is reduced and resolution is in-creased. Given the improvements and the modest computational cost, many devices like webcams will employ bilinear interpolation to reconstruct the image.

Variable number of Gradients

= VNG

This CFA recovery algorithm uses a threshold-based variable number of gradients. In order to recover missing color information at each pixel, we measure the gradient in eight directions based on a 5 X 5 neighborhood surrounding that pixel. Each gradient value is defined as a linear combination of the absolute differences of the like-colored pixels in this neighborhood. We then consider the entire set of eight gradients to determine a threshold of acceptable gradients. For all of the gradients that pass the threshold test, we use color components from the corresponding areas of the 5 X 5 neighborhoods to determine the missing color values.

Color filter array recovery using a threshold-based variable number of gradients[p.200]

Adaptive Homogeneity-Directed

= AHD

The direction to interpolate is chosen to minimize the level of color artifacts.

Due to the rectangular sampling lattice in Bayer pattern, interpolation is performed in horizontal and vertical directions only .

Directional interpolation uses filterbank techniques to cancel aliasing from CFA sampling.

Adaptive Homogeneity-directed Demosaicing[p.200]

Patterned Pixel Grouping

= PPG

Patterned Pixel Grouping interpolation is an advanced interpolation method which is designed to maintain the high image quality and minimize the interpolation artifacts from the output.

Pixel Grouping[p.200]

1.12.1.3. Image Gamma Correction

Linear

No gamma correction is applied to the image.

BT.709 curve

Apply a specific gamma curve to the image.

BT.709

= Rec. 709

Standardizes the format of high-definition television, having 16:9 (widescreen) aspect ratio. The first edition of the standard was approved in 1990.

Recommendation ITU BT.709*[http://www.itu.int/rec/R-REC-BT.709/en]*

sRGB curve

Apply a specific gamma curve to the image.

sRGB*[https://www.w3.org/Graphics/Color/sRGB.html]*

1.12.1.4. Image White Balance

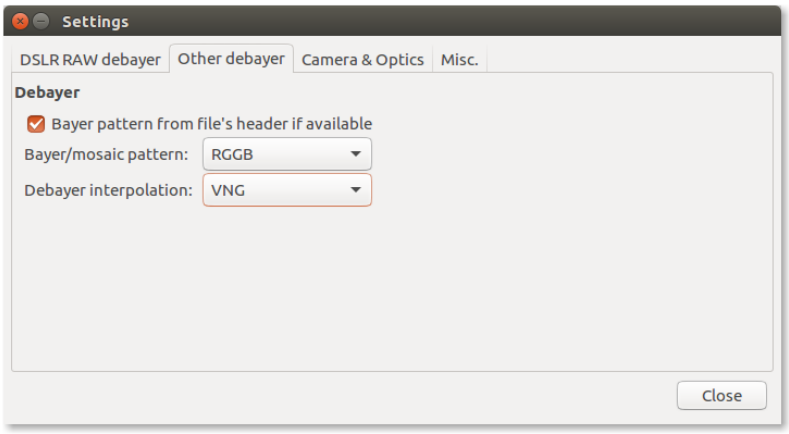
Use the white balance specified by the camera

By default, libraw uses a fixed white balance based on a color chart illuminated with a standard D65 lamp.*[https://en.wikipedia.org/wiki/Illuminant_D65]* If you check this box, libraw will **try to use** data specified by your camera.

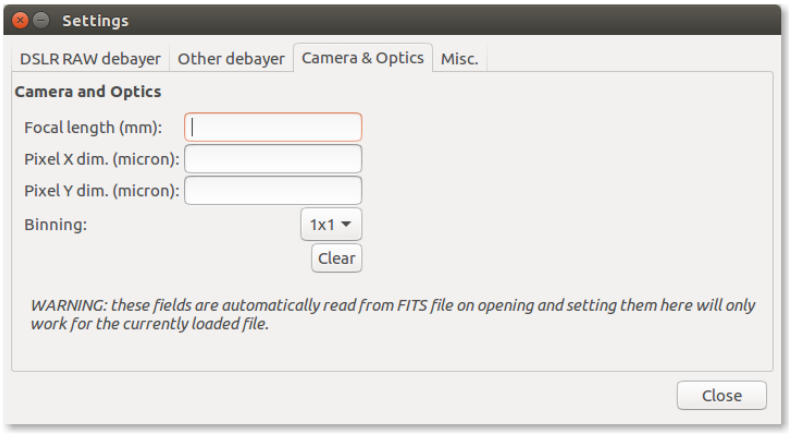
Calculate the white balance by averaging the entire image

By default, libraw uses a fixed white balance based on a color chart illuminated with a standard D65 lamp. If you check this box, libraw will **compute** the white balance by averaging the entire image.

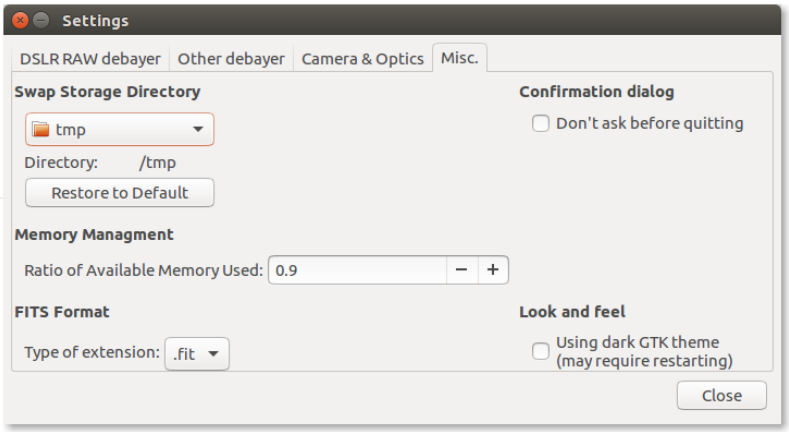
1.12.2. Other debayer



1.12.3. Camera & Optics



1.12.4. Misc.



1.13. Contextual help

On mouseover, context sensitive help is available for most input box, radio button, checkbox button text.

Allow a few seconds motionless mouse to access a tooltip.

1.14. Keyboard shortcuts

Ctrl+O

Open file

Ctrl+S

Saving the image (FIT format)

Ctrl+T

Saving the image (TIFF format)

Ctrl+B

Saving the BMP image

Ctrl+J

Saving the JPG image

Ctrl+P

Saving the NetPBM image

Ctrl+Z

Undo

Shift+Ctrl+Z

Redo

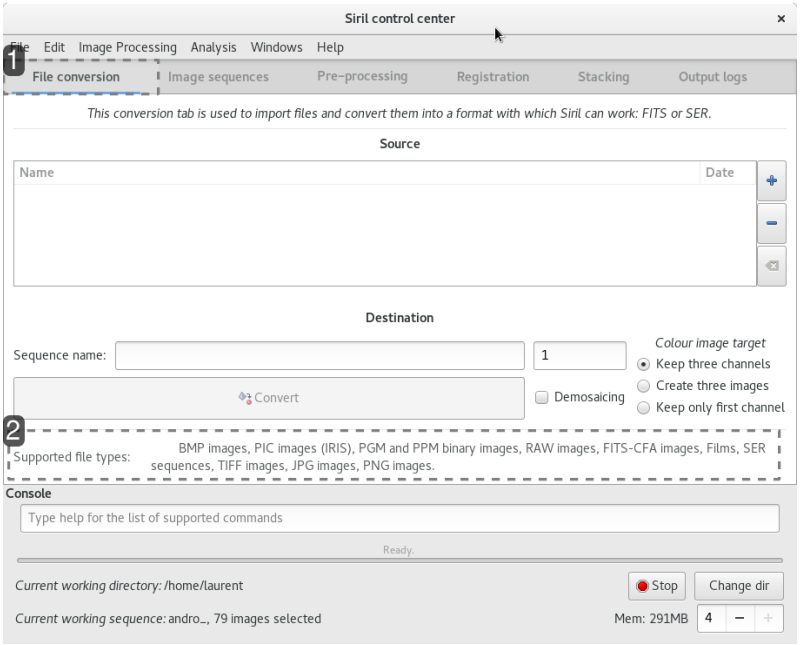
2. Tutorials

2.1. Deep Sky

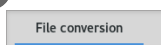
2.1.1. Convert your images in the FITS format Siril uses (image import)

To process your images with Siril, you must convert them to the FITS[https://fr.wikipedia.org/wiki/Flexible_Image_Transport_System] format it uses (16-bit unsigned, bottom-top order, 1 or 3 axes).

Fortunately, Siril is able to convert some image formats to this format, RAW, TIFF, JPEG, PIC[p.200], PNG, BMP and NetPBM binaries for images, AVI and other films.

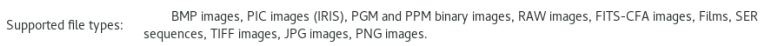


1



conversion tab

2


Supported file types:
BMP images, PIC images (IRIS), PGM and PPM binary images, RAW images, FITS-CFA images, Films, SER sequences, TIFF images, JPG images, PNG images.

Supported Formats

when pretreated images do not directly demosaicing RAW files !

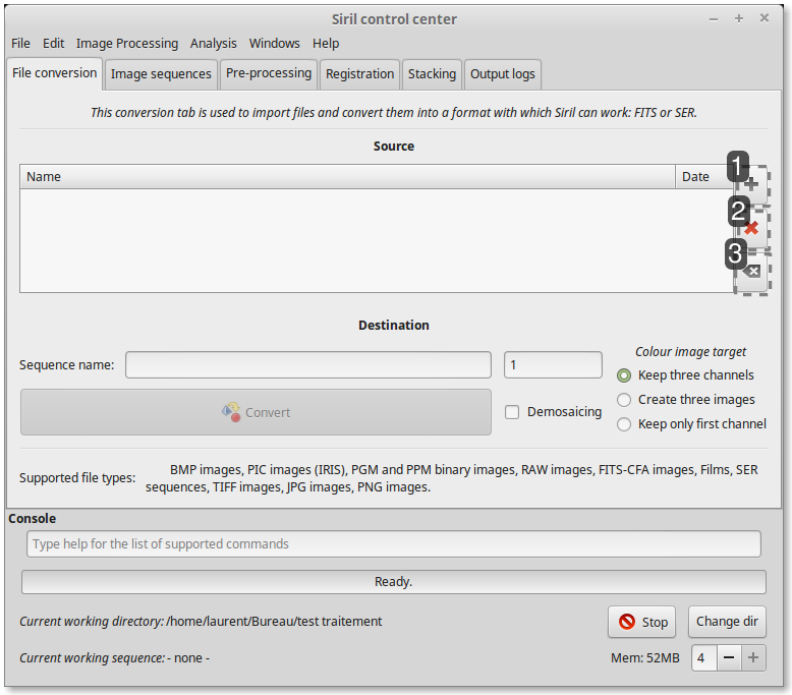
2.1.1.1. Images formats natively supported

- FITS images in 16-bit unsigned (fits other formats are converted on the fly)
- SER files
- AVI and many other films.

The AVI support as has been abandoned in favor of **SER** format. Consequently, several functions may not work with AVI movies. However, this version of Siril can convert AVI movies in SER format which will then be perfectly usable.

2.1.1.2. add the files to be converted

Click on the + button to add the files to be converted.



1 +

Add the files to convert

2

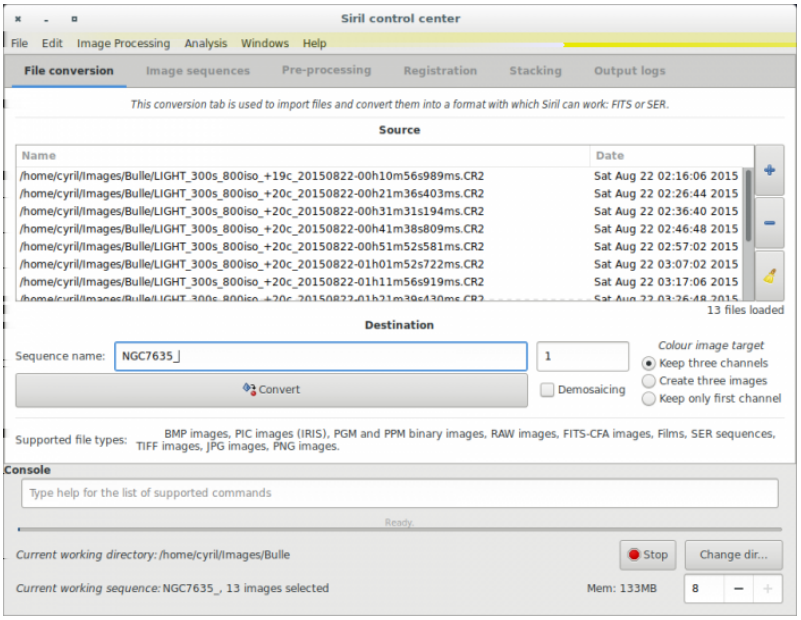
Remove files

3

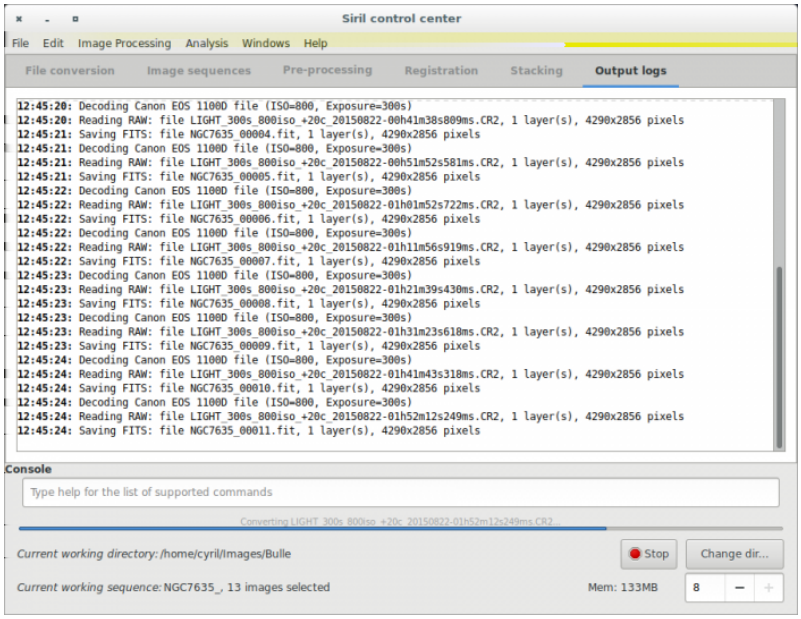
Remove all files

All supported formats can be selected at same time and will be converted in a sequence of files starting with the generic name you've specified just below. Created file names will be in the form Name_XXX, where XXX is the number of each image in the sequence.

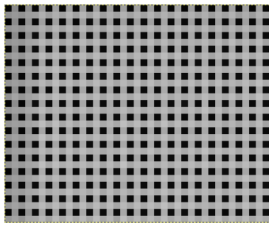
The button ☑Demosaicing must be unchecked in order to convert RAW files into CFA (Color Filter Array[*https://en.wikipedia.org/wiki/Color_filter_array*]). monochrome FITS pictures, a necessary step to pre-process **DSLR images**.


Siril control center
File Edit Image Processing Analysis Windows Help
File conversion
Image sequences
Pre-processing
Registration
Stacking
Output logs
This conversion tab is used to import files and convert them into a format with which Siril can work: FITS or SER.
Source
Name
Date
/home/cyril/Images/Bulle/LIGHT_300s_800iso_+19c_20150822-00h10m56s989ms.CR2 Sat Aug 22 02:16:06 2015
/home/cyril/Images/Bulle/LIGHT_300s_800iso_+20c_20150822-00h21m36s403ms.CR2 Sat Aug 22 02:26:44 2015
/home/cyril/Images/Bulle/LIGHT_300s_800iso_+20c_20150822-00h31m31s194ms.CR2 Sat Aug 22 02:36:40 2015
/home/cyril/Images/Bulle/LIGHT_300s_800iso_+20c_20150822-00h41m38s809ms.CR2 Sat Aug 22 02:46:48 2015
/home/cyril/Images/Bulle/LIGHT_300s_800iso_+20c_20150822-00h51m52s581ms.CR2 Sat Aug 22 02:57:02 2015
/home/cyril/Images/Bulle/LIGHT_300s_800iso_+20c_20150822-01h01m52s722ms.CR2 Sat Aug 22 03:07:02 2015
/home/cyril/Images/Bulle/LIGHT_300s_800iso_+20c_20150822-01h11m56s919ms.CR2 Sat Aug 22 03:17:06 2015
13 files loaded
Destination
Sequence name: NGC7635_
1
Convert
Demosaicing
Colour image target
Keep three channels
Create three images
Keep only first channel
Supported file types: BMP images, PIC images (IRIS), PGM and PPM binary images, RAW images, FITS-CFA images, Films, SER sequences, TIFF images, JPG images, PNG images.
Console
Type help for the list of supported commands
Ready.
Current working directory: /home/cyril/Images/Bulle
Stop
Change dir...
Current working sequence: NGC7635_, 13 images selected
Mem: 133MB
8

logs


Siril control center
File Edit Image Processing Analysis Windows Help
File conversion
Image sequences
Pre-processing
Registration
Stacking
Output logs
12:45:20: Decoding Canon EOS 1100D file (ISO=800, Exposure=300s)
12:45:20: Reading RAW: file LIGHT_300s_800iso_+20c_20150822-00h41m38s809ms.CR2, 1 layer(s), 4290x2856 pixels
12:45:21: Saving FITS: file NGC7635_00004.fit, 1 layer(s), 4290x2856 pixels
12:45:21: Decoding Canon EOS 1100D file (ISO=800, Exposure=300s)
12:45:21: Reading RAW: file LIGHT_300s_800iso_+20c_20150822-00h51m52s581ms.CR2, 1 layer(s), 4290x2856 pixels
12:45:21: Saving FITS: file NGC7635_00005.fit, 1 layer(s), 4290x2856 pixels
12:45:22: Decoding Canon EOS 1100D file (ISO=800, Exposure=300s)
12:45:22: Reading RAW: file LIGHT_300s_800iso_+20c_20150822-01h01m52s722ms.CR2, 1 layer(s), 4290x2856 pixels
12:45:22: Saving FITS: file NGC7635_00006.fit, 1 layer(s), 4290x2856 pixels
12:45:22: Decoding Canon EOS 1100D file (ISO=800, Exposure=300s)
12:45:22: Reading RAW: file LIGHT_300s_800iso_+20c_20150822-01h11m56s919ms.CR2, 1 layer(s), 4290x2856 pixels
12:45:22: Saving FITS: file NGC7635_00007.fit, 1 layer(s), 4290x2856 pixels
12:45:23: Decoding Canon EOS 1100D file (ISO=800, Exposure=300s)
12:45:23: Reading RAW: file LIGHT_300s_800iso_+20c_20150822-01h21m39s430ms.CR2, 1 layer(s), 4290x2856 pixels
12:45:23: Saving FITS: file NGC7635_00008.fit, 1 layer(s), 4290x2856 pixels
12:45:23: Decoding Canon EOS 1100D file (ISO=800, Exposure=300s)
12:45:23: Reading RAW: file LIGHT_300s_800iso_+20c_20150822-01h31m23s618ms.CR2, 1 layer(s), 4290x2856 pixels
12:45:23: Saving FITS: file NGC7635_00009.fit, 1 layer(s), 4290x2856 pixels
12:45:24: Decoding Canon EOS 1100D file (ISO=800, Exposure=300s)
12:45:24: Reading RAW: file LIGHT_300s_800iso_+20c_20150822-01h41m43s318ms.CR2, 1 layer(s), 4290x2856 pixels
12:45:24: Saving FITS: file NGC7635_00010.fit, 1 layer(s), 4290x2856 pixels
12:45:24: Decoding Canon EOS 1100D file (ISO=800, Exposure=300s)
12:45:24: Reading RAW: file LIGHT_300s_800iso_+20c_20150822-01h52m12s249ms.CR2, 1 layer(s), 4290x2856 pixels
12:45:24: Saving FITS: file NGC7635_00011.fit, 1 layer(s), 4290x2856 pixels
Console
Type help for the list of supported commands
Converting LIGHT_300s_800iso_+20c_20150822-01h52m12s249ms.CR2
Current working directory: /home/cyril/Images/Bulle
Stop
Change dir...
Current working sequence: NGC7635_, 13 images selected
Mem: 133MB
8

Indeed, due to Bayer matrix consideration, the RGB result of your RAW image is an interpolated picture. In consequence pre-processing interpolated data will give wrong results. Contrary to RGB image, CFA image represent the entire sensor data with the Bayer pattern. The following image shows you a crop of a CFA image. Note that the Bayer pattern (RGGB on this example) is visible.



2.1.2. Pre-processing images

Once a sequence is loaded, images can be pre-processed, registered and stacked. The pre-processing is an optional, yet important, step and involves bias, dark and flat frames. Pre-processing a sequence in Siril can only be done with master bias, dark and flat frame, which have to be created from their sequences first. The way these master frames can be created is described in the sections below, see also the stacking documentation [p.38].

Pre-processing

The preprocessing operation consists in computing the following equation:

$$\textbf{calibrated image} = \frac{light - masterDark - masterBias}{masterFlat - masterFlatDark - masterBias}$$

For pre-processing, Siril applies master offset/bias, dark and flat images to the current sequence. These master images thus have to be processed before processing the actual image sequence. Siril currently does not support processing multiple sequences at the same time, so each layer of the final image must be processed independently before assembling them into an RGB image.

2.1.2.1. Prepare dark, flat and bias

2.1.2.1.a. Bias

A BIAS frame

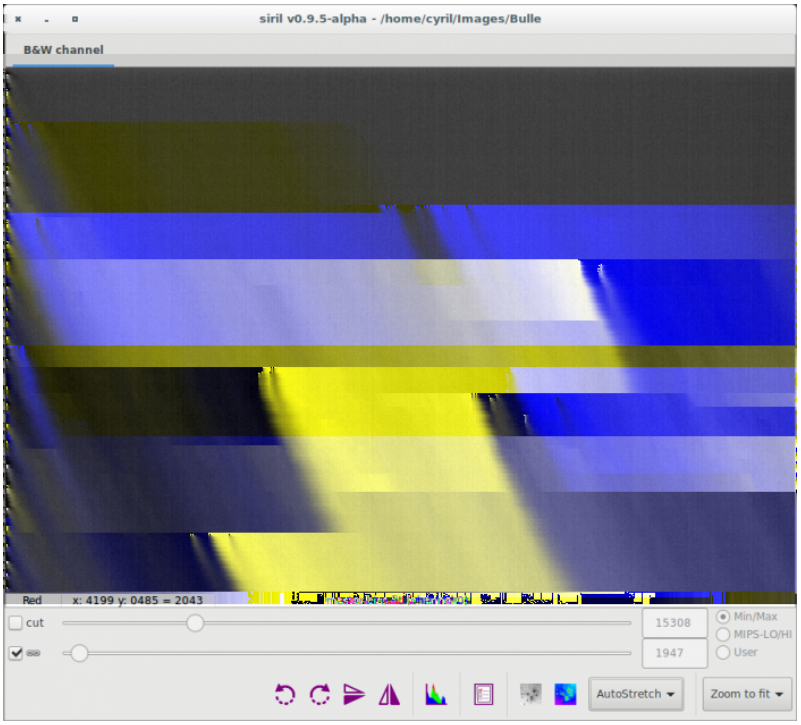
= offset

is an image taken with the shutter closed and the shortest possible exposure time. Basically it corresponds to an exposure of 1/4000s with modern DSLRs. The BIAS shows the electronic noise and readout signal of the camera and the more BIAS exposures are used for the calculation of the master BIAS, the less noise will be introduced into the corrected images. Note that BIAS frames MUST be taken at the same ISO than light frames.

The master bias frame should be created by stacking bias frames with the median algorithm (or Winsorized by checking the rejection levels at the end of the process, they should be lower than 0.5 percent), but be sure to use **No Normalization**.

Here, a master BIAS, calculated from 10 single images with Winsorized sigma clipping.

```
12:58:34: Pixel rejection in channel #0: 0.368% - 0.354%
12:58:34: Integration of 10 images:
12:58:34: Normalization ............. None
12:58:34: Pixel rejection ........... Winsorized sigma clipping
12:58:34: Rejection parameters ...... low=3.200 high=3.300
```



2.1.2.1.b. Dark

Darks

Dark frames are made at the same exposure time and ISO as the subject light frames but in the dark: use your lens/telescope cap or close the shutter for example.

They contain the thermal noise associated with the sensor, the noise being proportional to temperature and exposure time. Hence, they should be made at approximately the same temperature as the light frames, this is the reason why we make dark frames at the end, or in the middle of the imaging session.

Like with the BIAS frames, the more dark exposures are used for the calculation of the master dark, the less noise will be introduced into the corrected images.

The master dark frame should be created by stacking dark frames with the median algorithm (or Winsorized by checking the rejection levels at the end of the process, they should be lower than 0.5 percent), but be sure to use **No** 👉👉Normalisation

Remember that dark frames are always composed from real dark signal and bias signal.

If you don't apply dark optimization, you can leave the bias signal and your masterDark will be in fact masterDark+masterBias.

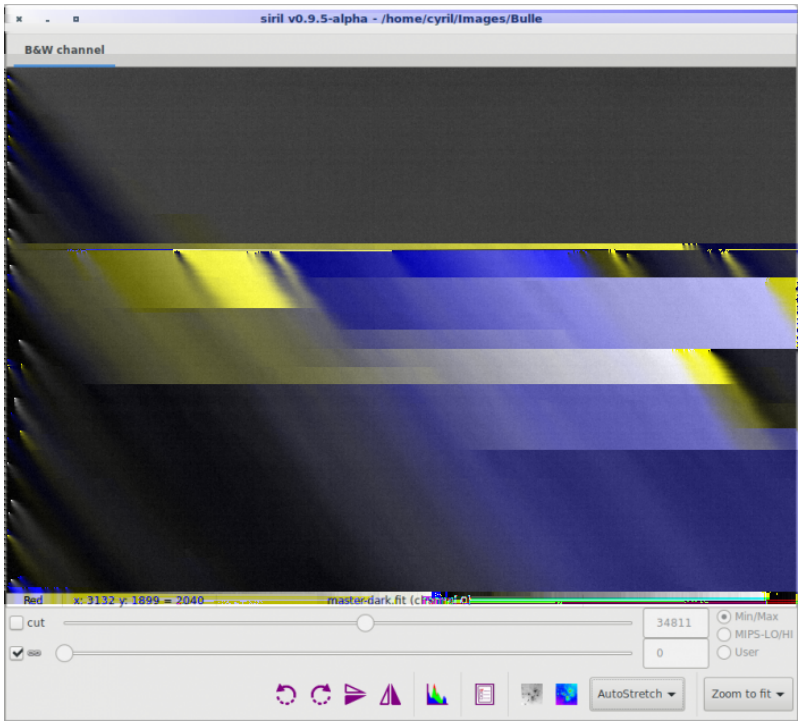
In consequence subtracting this master to the light frames will remove both signals.

However, applying dark optimization makes things different by multiplying masterDark by a coefficient factor not equal to 1.

In this case, you must subtract masterBias from each dark frame.

Here, a master dark, calculated from 10 single images with Winsorized sigma clipping.

```
13:00:40: Pixel rejection in channel #0: 0.340% - 0.384%
13:00:40: Integration of 10 images:
13:00:40: Normalization ............. None
13:00:40: Pixel rejection ........... Winsorized sigma clipping
13:00:40: Rejection parameters ...... low=3.200 high=3.300
```



2.1.2.1.c. Flat

Flat

Telescopes usually do not illuminate the detector homogeneously. Moreover, dust on optical surfaces and sensor lead to darker patterns on resulting image, and the sensor itself has a different response to the number of photons hitting different photosites. To correct these effects one need to divide each light images by the master flat, which should be the median of single exposures of a homogeneously and non-saturating illuminated area.

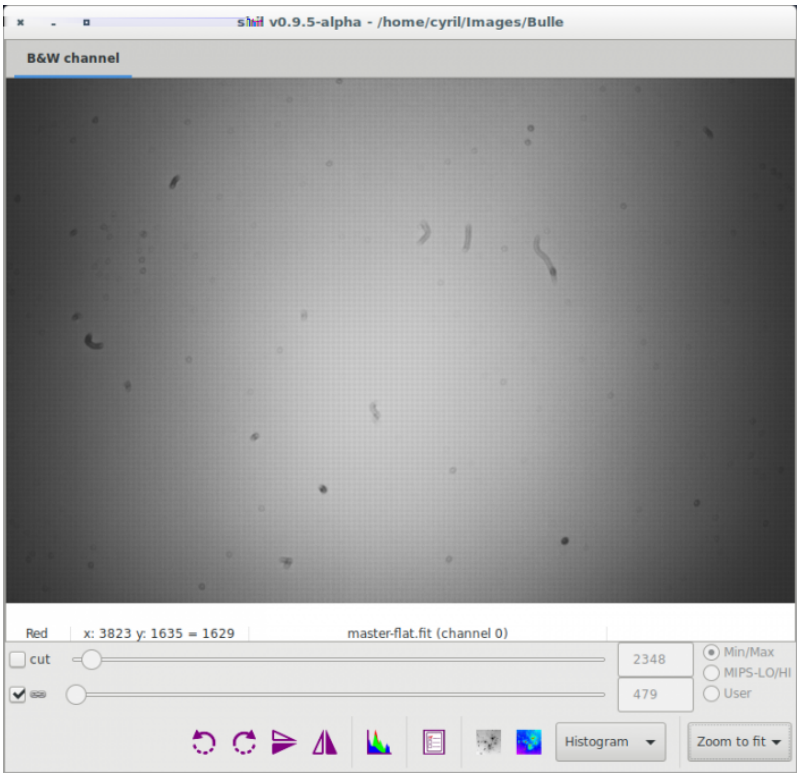
Remember that flat frames are always composed from real flat signal and bias signal. So you must subtract master-bias from each flat frames to obtain real master-flat. In addition, if flat frame exposures are too long, a dark-flat signal should also be removed.

Siril automatically evaluates the normalisation coefficient used in the flat division. However, this value can be self-evaluated.

The master flat frame should be created by stacking flat frames with the median algorithm (or Winsorized by checking the rejection levels at the end of the process, which should be lower than 0.500), and choose **Multiplicative normalisation**.

Here, a master flat, calculated from 9 single images with Winsorized sigma clipping method. Note that vignetting and dust are visible. Bayer pattern is also visible.

```
13:05:23: Pixel rejection in channel #0: 0.388% - 0.304%
13:05:23: Integration of 9 images:
13:05:23: Normalization ............. multiplicative
13:05:23: Pixel rejection ........... Winsorized sigma clipping
13:05:23: Rejection parameters ...... low=3.400 high=3.700
```



2.1.2.2. For DSLR images

Pre-processing DSLR images without dark optimization

When dark optimization is not used and masterFlatDark not needed, one good method is to process like that:

1. Integrate all bias to build master-bias
2. Integrate all darks to build master-dark
3. Calibrate flat frames with master-bias

4. Integrate all calibrated flat to build master-flat
5. Preprocess light frame with master-dark and master-flat

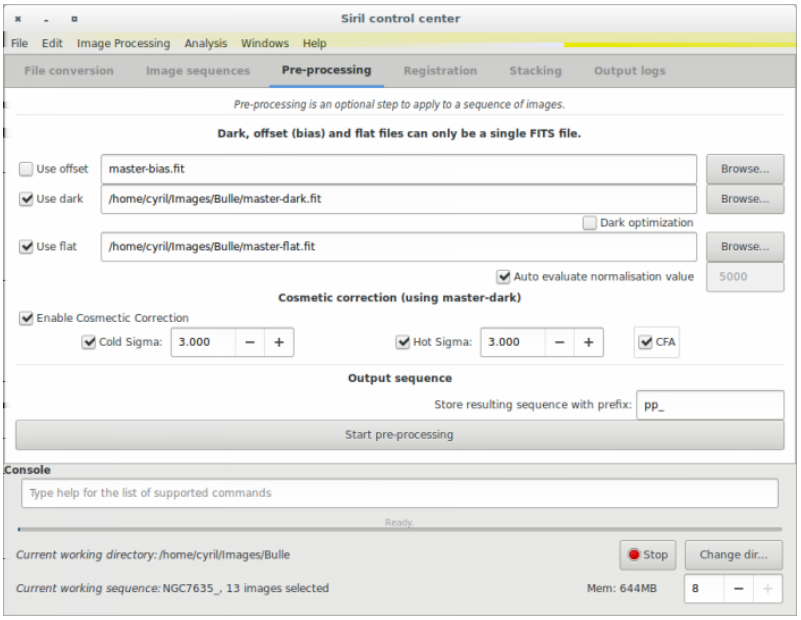
Whole DSLR images processing batch

> If you want to do the whole processing batch, once again with ***no masterFlatDark*** but **with dark optimization**, you must remove the ***master-bias*** from the ***master-dark*** before:

1. Integrate all bias to build master-bias
2. Calibrate dark frames with master-bias
3. Integrate all calibrated darks to build master-dark
4. Calibrate flat frames with master-bias
5. Integrate all calibrated flat to build master-flat
6. Preprocess light frame with master-bias, master-dark and master-flat

2.1.2.3. Pre-processing and FIT images

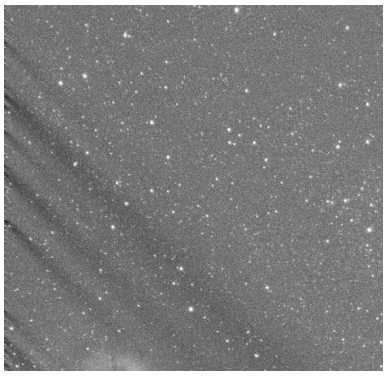
> The pre-processing does not overwrites original FITS files from the sequence but creates new FITS files with the pre-processed data.
>
> The new file names are the same than the original prefixed with pp_ if you leave the default value in the prefix box.
>
> Below is an example of pre-processing using offset and dark files.
>
> Make sure that loaded sequence is the sequence of your light frames.
>
> Also, note that at the end of the process, the new sequence is automatically loaded.



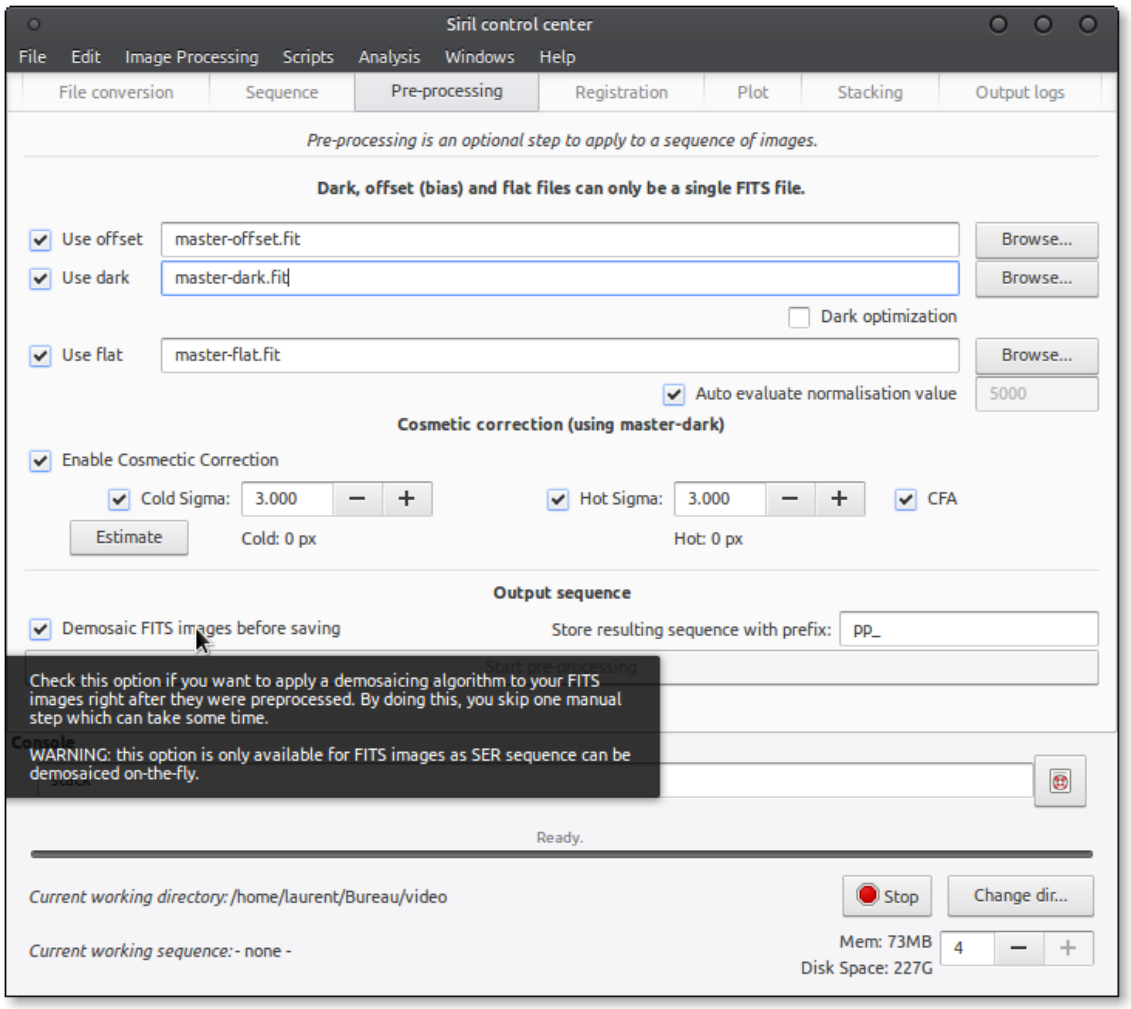
Since 0.9.2 version, a new tool is available in the preprocessing tab: cosmetic correction. This tool search deviant pixels through master-dark and will correct them. You always should use this tool if you own master-dark. Please, pay attention to the CFA box. Check it if your monochrome pictures are CFA images. If you don't have master-dark (which should be avoided), you could use automatic cosmetic correction tool available through the image processing menu.

Results examples

The result of a single frame pre-processed is given below. Note how dust have been removed from the light frame.



2.1.2.3.a. Simplified workflow since 0.9.9


Siril control center
File Edit Image Processing Scripts Analysis Windows Help
File conversion Sequence Pre-processing Registration Plot Stacking Output logs
Pre-processing is an optional step to apply to a sequence of images.
Dark, offset (bias) and flat files can only be a single FITS file.
Use offset master-offset.fit Browse...
Use dark master-dark.fit Browse...
Dark optimization
Use flat master-flat.fit Browse...
Auto evaluate normalisation value 5000
Cosmetic correction (using master-dark)
Enable Cosmectic Correction
Cold Sigma: 3.000
Hot Sigma: 3.000
CFA
Estimate
Cold: 0 px
Hot: 0 px
Output sequence
Demosaic FITS images before saving
Store resulting sequence with prefix: pp_
Check this option if you want to apply a demosaicing algorithm to your FITS images right after they were preprocessed. By doing this, you skip one manual step which can take some time.
WARNING: this option is only available for FITS images as SER sequence can be demosaiced on-the-fly.
Console
Ready.
Current working directory: /home/laurent/Bureau/video
Stop
Change dir...
Current working sequence: - none -
Mem: 73MB
Disk Space: 227G
4

You can check ☑demosaic FTIS images option before saving.
This allows you to skip the manual conversion step

Valid only for FITS, Ser files are demosaiced on the fly.

2.1.3. Special case of demosaicing

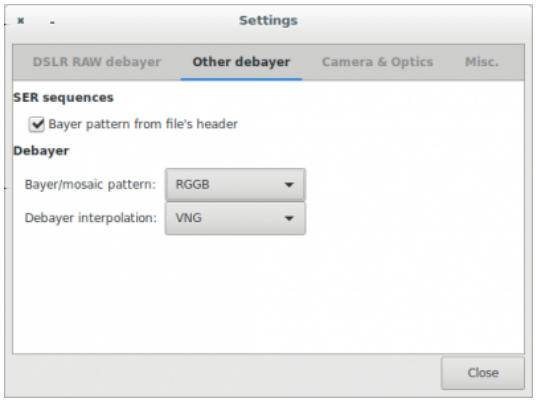
Demosaicing (for Bayer-Pattern images only)

Last step in pre-processing images consist in getting back Red, Green and Blue channels using interpolation algorithms. To do that, Siril gives the choice between 5 different algorithms.

- VNG (default)
- Bilinear (low quality but fast)

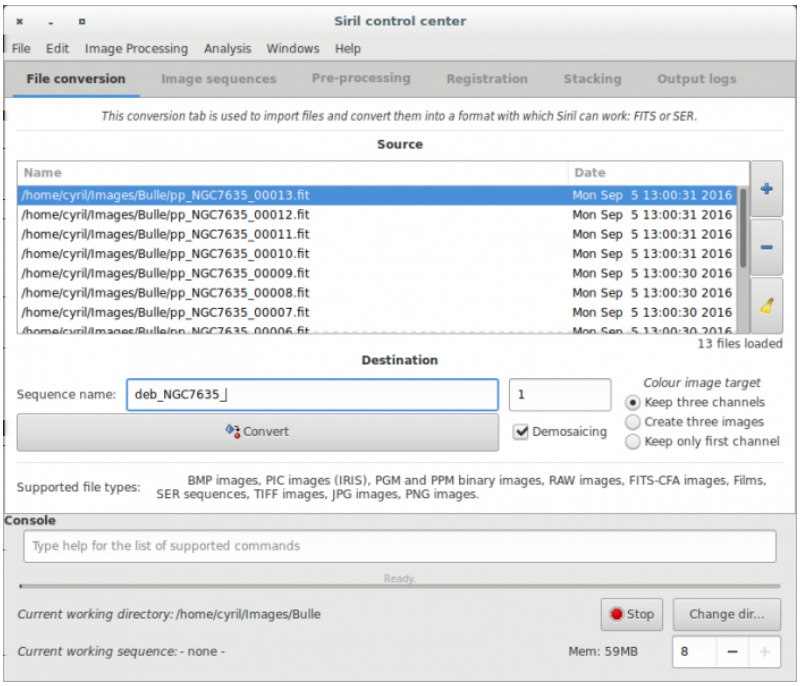
- Nearest neighboor
- AHD
- Super Pixel (no interpolation, the pattern RGGB, or other, is used as a super pixel and the image size is divided by four).

Make your choice in the window below by clicking on Settings and Other debayer tab.

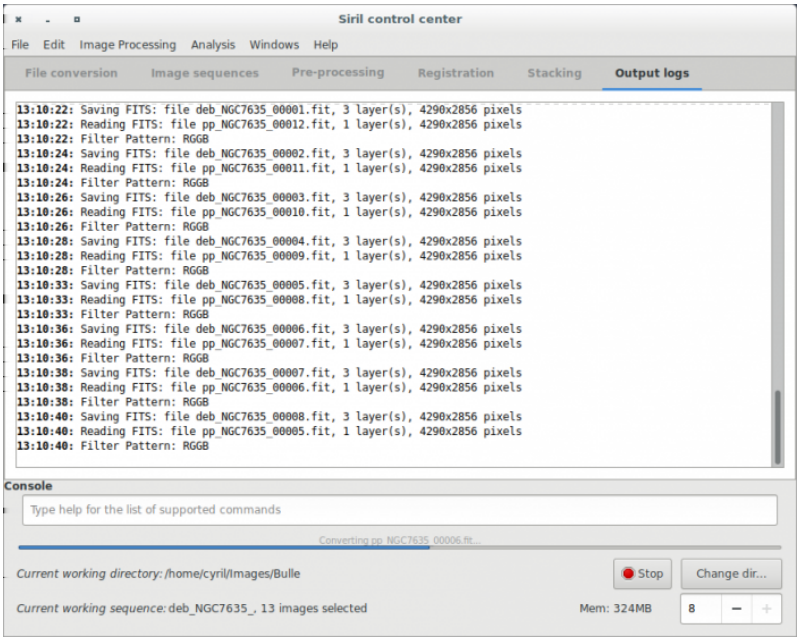


To process the demosaicing

1. Go back to Conversion Tab and fill the field as the example below:


Siril control center
File Edit Image Processing Analysis Windows Help
File conversion
Image sequences
Pre-processing
Registration
Stacking
Output logs
This conversion tab is used to import files and convert them into a format with which Siril can work: FITS or SER.
Source
Name
Date
/home/cyril/Images/Bulle/pp_NGC7635_00013.fit Mon Sep 5 13:00:31 2016
/home/cyril/Images/Bulle/pp_NGC7635_00012.fit Mon Sep 5 13:00:31 2016
/home/cyril/Images/Bulle/pp_NGC7635_00011.fit Mon Sep 5 13:00:31 2016
/home/cyril/Images/Bulle/pp_NGC7635_00010.fit Mon Sep 5 13:00:31 2016
/home/cyril/Images/Bulle/pp_NGC7635_00009.fit Mon Sep 5 13:00:30 2016
/home/cyril/Images/Bulle/pp_NGC7635_00008.fit Mon Sep 5 13:00:30 2016
/home/cyril/Images/Bulle/pp_NGC7635_00007.fit Mon Sep 5 13:00:30 2016
13 files loaded
Destination
Sequence name: deb_NGC7635_
1
Convert
Demosaicing
Colour image target
Keep three channels
Create three images
Keep only first channel
Supported file types: BMP images, PIC images (IRIS), PGM and PPM binary images, RAW images, FITS-CFA images, Films, SER sequences, TIFF images, JPG images, PNG images.
Console
Type help for the list of supported commands
Ready.
Current working directory: /home/cyril/Images/Bulle
Stop
Change dir...
Current working sequence: - none -
Mem: 59MB
8

2. Then check the Demosaicing button and click on the Convert Button right after. The demosaicing process should start and will take some time following the method you used and the CPU you own.

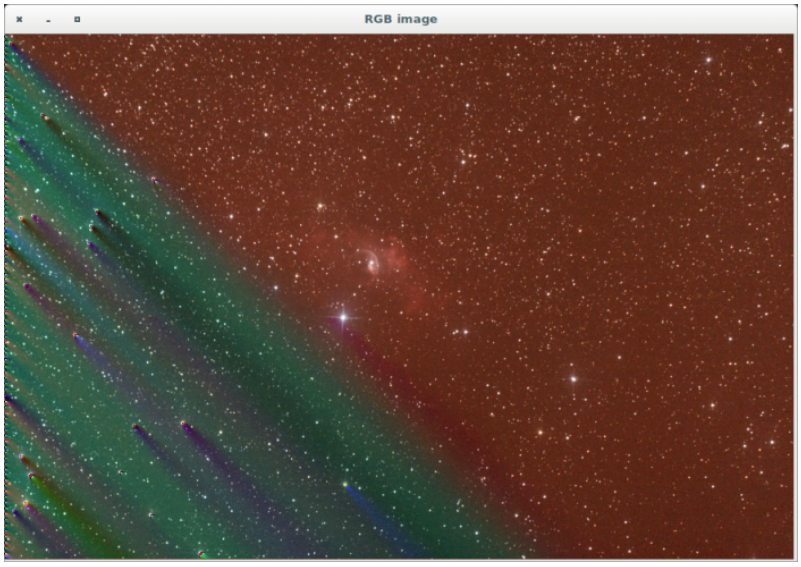

Siril control center
File Edit Image Processing Analysis Windows Help
File conversion
Image sequences
Pre-processing
Registration
Stacking
Output logs
13:10:22: Saving FITS: file deb_NGC7635_00001.fit, 3 layer(s), 4290x2856 pixels
13:10:22: Reading FITS: file pp_NGC7635_00012.fit, 1 layer(s), 4290x2856 pixels
13:10:22: Filter Pattern: RGGB
13:10:24: Saving FITS: file deb_NGC7635_00002.fit, 3 layer(s), 4290x2856 pixels
13:10:24: Reading FITS: file pp_NGC7635_00011.fit, 1 layer(s), 4290x2856 pixels
13:10:24: Filter Pattern: RGGB
13:10:26: Saving FITS: file deb_NGC7635_00003.fit, 3 layer(s), 4290x2856 pixels
13:10:26: Reading FITS: file pp_NGC7635_00010.fit, 1 layer(s), 4290x2856 pixels
13:10:26: Filter Pattern: RGGB
13:10:28: Saving FITS: file deb_NGC7635_00004.fit, 3 layer(s), 4290x2856 pixels
13:10:28: Reading FITS: file pp_NGC7635_00009.fit, 1 layer(s), 4290x2856 pixels
13:10:28: Filter Pattern: RGGB
13:10:33: Saving FITS: file deb_NGC7635_00005.fit, 3 layer(s), 4290x2856 pixels
13:10:33: Reading FITS: file pp_NGC7635_00008.fit, 1 layer(s), 4290x2856 pixels
13:10:33: Filter Pattern: RGGB
13:10:36: Saving FITS: file deb_NGC7635_00006.fit, 3 layer(s), 4290x2856 pixels
13:10:36: Reading FITS: file pp_NGC7635_00007.fit, 1 layer(s), 4290x2856 pixels
13:10:36: Filter Pattern: RGGB
13:10:38: Saving FITS: file deb_NGC7635_00007.fit, 3 layer(s), 4290x2856 pixels
13:10:38: Reading FITS: file pp_NGC7635_00006.fit, 1 layer(s), 4290x2856 pixels
13:10:38: Filter Pattern: RGGB
13:10:40: Saving FITS: file deb_NGC7635_00008.fit, 3 layer(s), 4290x2856 pixels
13:10:40: Reading FITS: file pp_NGC7635_00005.fit, 1 layer(s), 4290x2856 pixels
13:10:40: Filter Pattern: RGGB
Console
Type help for the list of supported commands
Converting pp_NGC7635_00006.fit...
Current working directory: /home/cyril/Images/Bulle
Stop
Change dir...
Current working sequence: deb_NGC7635_, 13 images selected
Mem: 324MB
8

Result

Finally you will have a new set of pre-processed light frames, automatically loaded in Siril, that you could register.

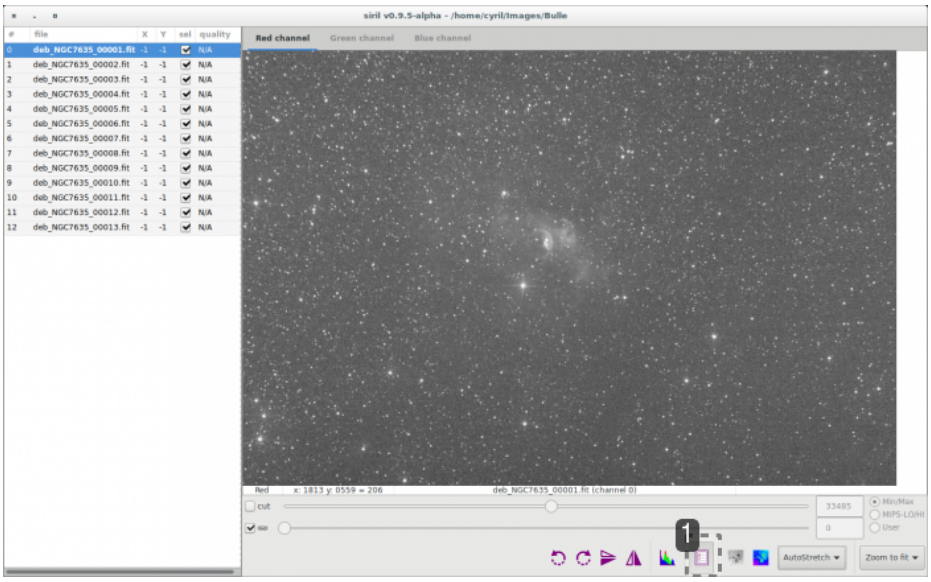
Please note that at this stage you always have a lot of noise in your image.

> Indeed, pre-processing images with Dark, Bias and Flat do not remove noise but unwanted signal.
>
> The following Picture shows you the result of demosaicing.
>
> It is displayed in Auto-stretch visualisation mode in order to enhanced the visibility of signal but without altering your image.



2.1.4. Use of registration

> After debayering (if useful), the new sequence is loaded and you can check it in the `image sequences` tab.
>
> From this tab, you can change the image being viewed in the grey and color windows, include or exclude them from a user-defined selection that will later be used to indicate which files should be stacked or not, and set the reference image.
>
> While browsing through the images of the sequence, a reference image may be set to a specific image if you think it has a better quality than the others, or if it's best centred. Note that when the reference image is set, it will be saved in the sequence and it will be opened first when loading the sequence the following time.



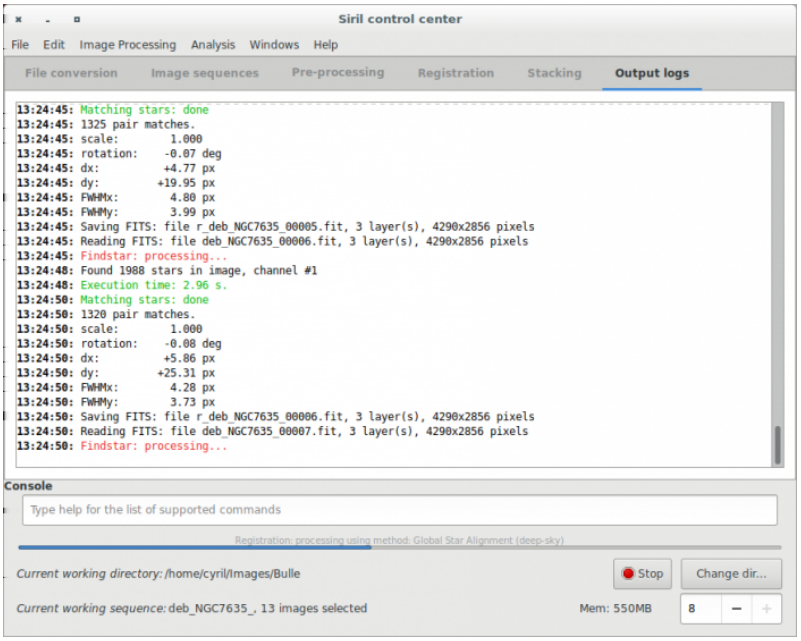
1

An alternative to the use of buttons of the sequence tab is to use the sequence list. Click on the dedicated button

We align the images on the best channels. Usually it is green.
Others will actually follow this alignment.
At the end it is possible to do an RGB alignment [p.173] if we have an offset.

To make registration, just click on Go Register button. Everything will be automatic.
The algorithm tries to match triangle patterns in order to align images.

You can check each step of registration in the console output.


Siril control center
File Edit Image Processing Analysis Windows Help
File conversion
Image sequences
Pre-processing
Registration
Stacking
Output logs
13:24:45: Matching stars: done
13:24:45: 1325 pair matches.
13:24:45: scale: 1.000
13:24:45: rotation: -0.07 deg
13:24:45: dx: +4.77 px
13:24:45: dy: +19.95 px
13:24:45: FWHMx: 4.80 px
13:24:45: FWHMy: 3.99 px
13:24:45: Saving FITS: file r_deb_NGC7635_00005.fit, 3 layer(s), 4290x2856 pixels
13:24:45: Reading FITS: file deb_NGC7635_00006.fit, 3 layer(s), 4290x2856 pixels
13:24:45: Findstar: processing...
13:24:48: Found 1988 stars in image, channel #1
13:24:48: Execution time: 2.96 s.
13:24:50: Matching stars: done
13:24:50: 1320 pair matches.
13:24:50: scale: 1.000
13:24:50: rotation: -0.08 deg
13:24:50: dx: +5.86 px
13:24:50: dy: +25.31 px
13:24:50: FWHMx: 4.28 px
13:24:50: FWHMy: 3.73 px
13:24:50: Saving FITS: file r_deb_NGC7635_00006.fit, 3 layer(s), 4290x2856 pixels
13:24:50: Reading FITS: file deb_NGC7635_00007.fit, 3 layer(s), 4290x2856 pixels
13:24:50: Findstar: processing...
Console
Type help for the list of supported commands
Registration: processing using method: Global Star Alignment (deep-sky)
Current working directory: /home/cyril/Images/Bulle
Stop
Change dir...
Current working sequence: deb_NGC7635_, 13 images selected
Mem: 550MB
8

A new sequence will be created and automatically loaded at the end of the process. In addition, an average of **FWHM** is computed and provided in this sequence. It is thus possible to select best images for stacking.

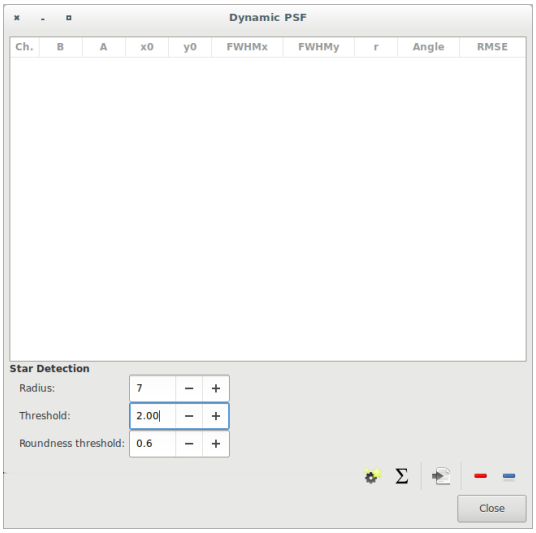
Information displayed in the `Output logs` tab

```
11:24:32: Findstar: processing...
11:24:32: Found 113 stars in image, channel #0
11:24:32: Execution time: 161.35 ms.
11:24:32: Matching stars: done
11:24:32: 62 pair matches.
11:24:32: scale: 1.000
11:24:32: rotation: -0.02 deg
11:24:32: dx: -7.18 px
11:24:32: dy: +3.30 px
11:24:32: FWHMx: 2.98 px
11:24:32: FWHMy: 2.54 px
11:24:32: Saving FITS: file r_ring_00015.fit, 1 layer(s), 1296x960 pixels
11:24:32: Reading FITS: file ring_00016.fit, 1 layer(s), 1296x960 pixels
```

Special cases with low sigma/noise ratio

In some cases, due to high noise level, automatic registration gives wrong results by confusing noise and stars during star detection. To overcome this issue, Siril provides adjustable parameters offering

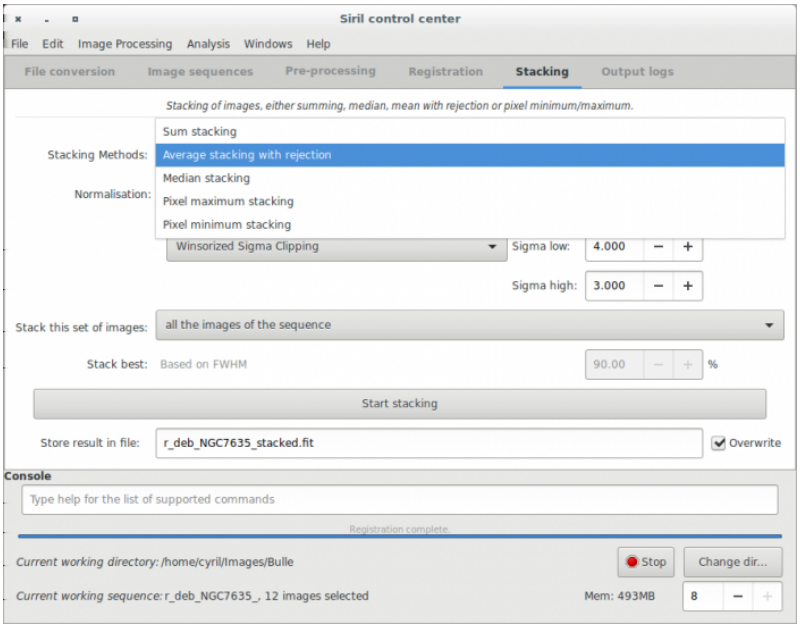
a better tool detection. These parameters are located in the Dynamic PSF window.



In this case, increasing threshold value will give better results.

2.1.5. Stacking

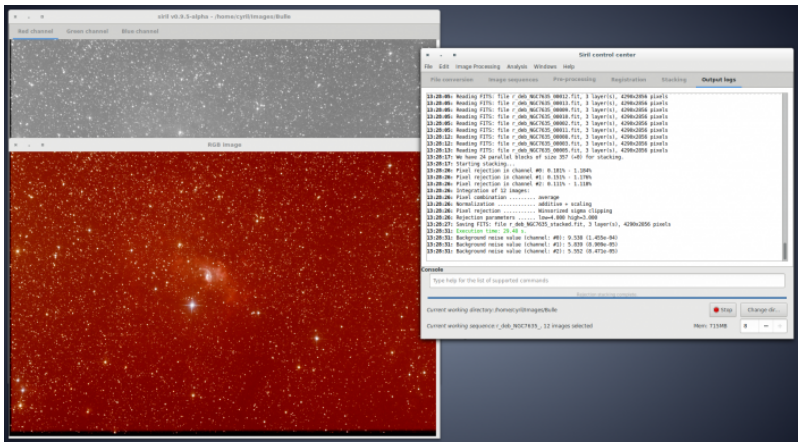
In the case of NGC7635 sequence, we first used the *Winsorized Sigma Clipping* algorithm in *Average stacking with rejection* section, in order to remove satellite tracks (σ_{low}=4 and σ_{high}=3).



The output console thus gives the following result:

```
22:26:06: Pixel rejection in channel #0: 0.215% - 1.401%
22:26:06: Pixel rejection in channel #1: 0.185% - 1.273%
22:26:06: Pixel rejection in channel #2: 0.133% - 1.150%
22:26:06: Integration of 12 images:
```

```
22:26:06: Normalization ..... additive + scaling
22:26:06: Pixel rejection ..... Winsorized sigma clipping
22:26:06: Rejection parameters ..... low=4.000 high=3.000
22:26:09: Saving FITS: file NGC7635.fit, 3 layer(s), 4290x2856 pixels
22:26:19: Background noise value (channel: #0): 10.013 (1.528e-04)
22:26:19: Background noise value (channel: #1): 6.755 (1.031e-04)
22:26:19: Background noise value (channel: #2): 6.621 (1.010e-04)
```



After that, the result is saved in the file named below the buttons, and is displayed in the grey and color windows.

You can adjust levels if you want to see it better, or use the different display mode. In our example the file is the stack result of all files, i.e., 12 files.

Now should start the process of the image

Start the process of the image with crop, background extraction (to remove gradient), and some other processes to enhance your image.

Here an example of what you can get with Siril

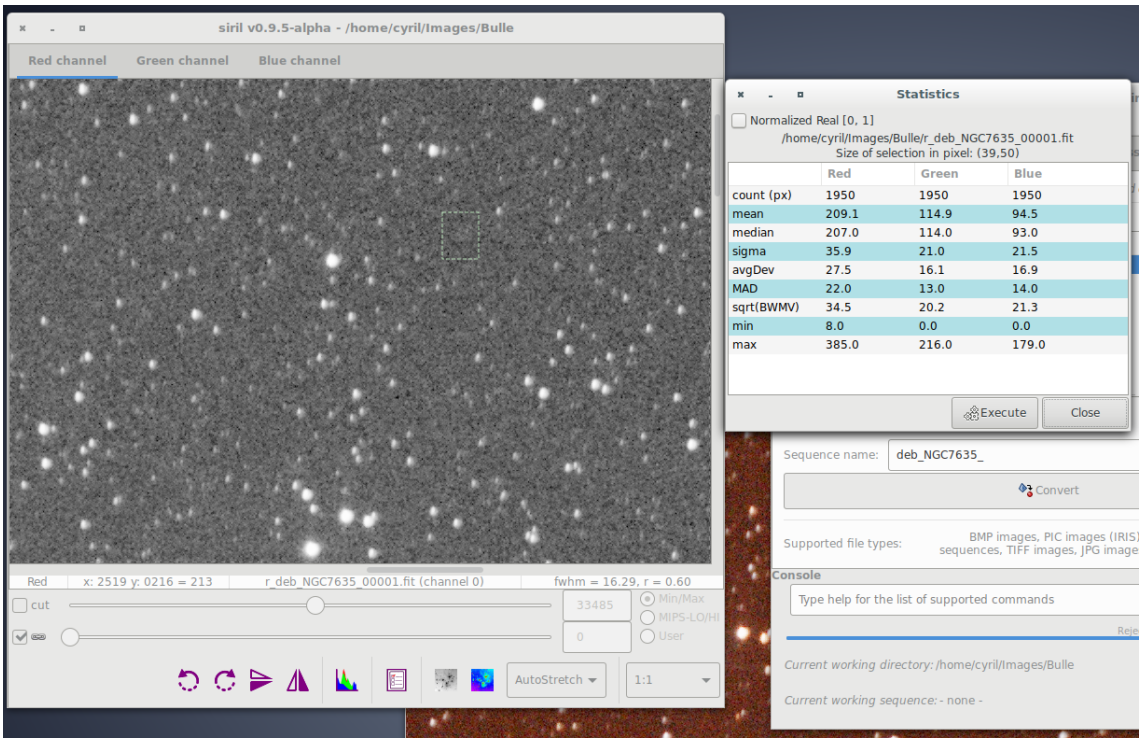
The purpose of the stack is the gain in signal to noise ratio.

Note the improvement of the signal-to-noise ratio regarding the result given between the first frame before stacking and the result of stacking.

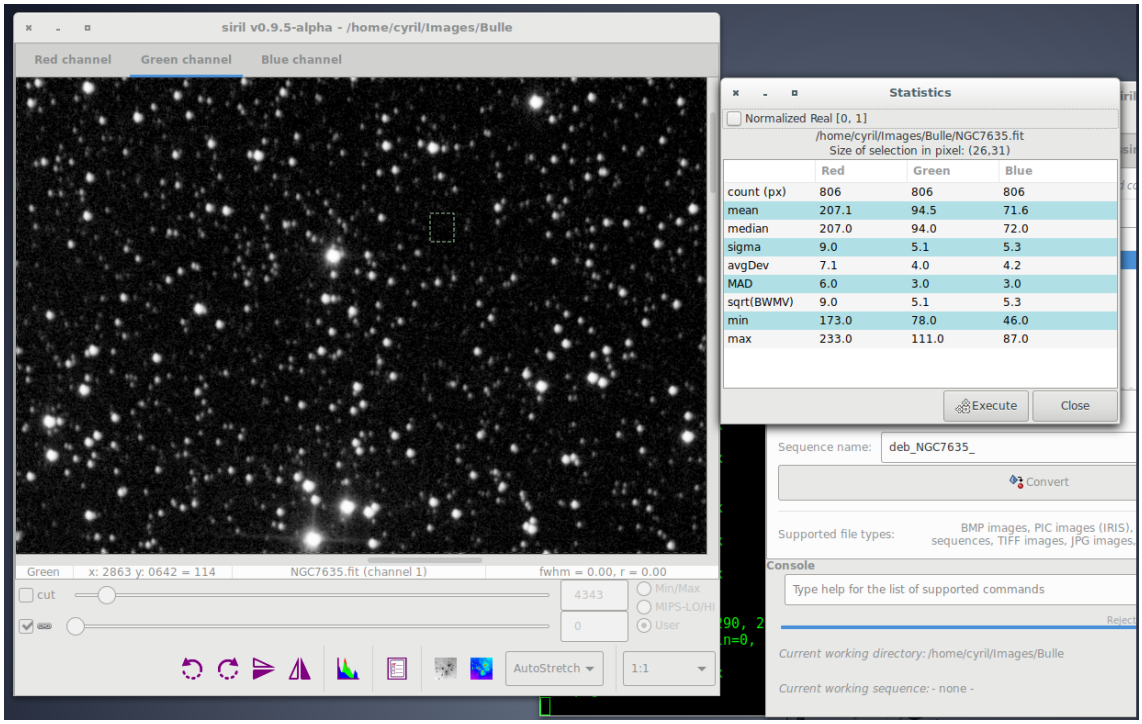
Take a look to the sigma value in the example below.

The increase in SNR is of 21/5.1=4.11≈1√12=3.46 and you should try to improve this kind of result adjusting σ_{low} and σ_{high}.

- first frame



- Stacking result



2.1.6. Process your single images

It is one thing to process a sequence of images, and another to work on only one image, generally the result of this processing. This section documents the most useful features for one-image processing.

Once you get your pre-processed and stacked image (fore more details see here), the process stage can start.

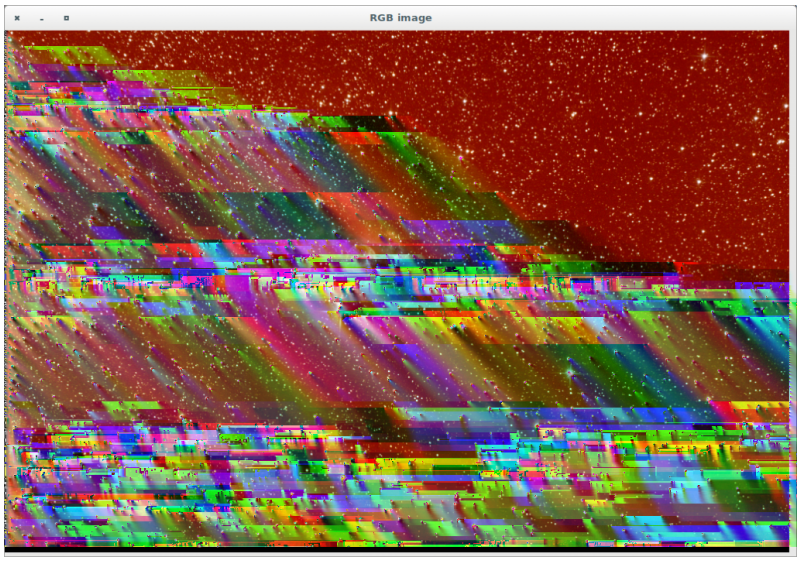
> These tutorials will present to you which different processes you can use, however there are no absolute rules. Feel free to try many processus by adjusting parameters.

For example a final result is illustrated in the picture below:

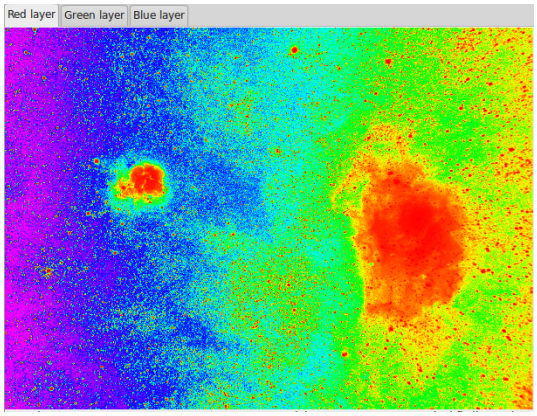
2.1.6.1. Background extraction

When an image contains an intense light pollution, it is useful to remove it. Siril provides a "Background Extraction" tool in order to remove the gradient of the sky background. Siril uses a polynomial up to the fourth degree. The 4th degree polynomial is generally a good choice, for this model, though the tool allows for first to fourth degree.

By selecting "Histogram Equalization" as display view, you can quickly check that image contains sky gradient due to light pollution, or in this case, Moon light.

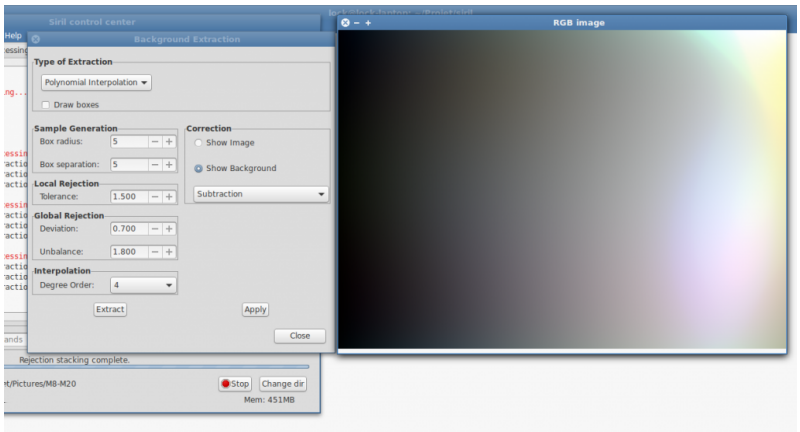


The rainbow colormap can also help you to see the gradient:



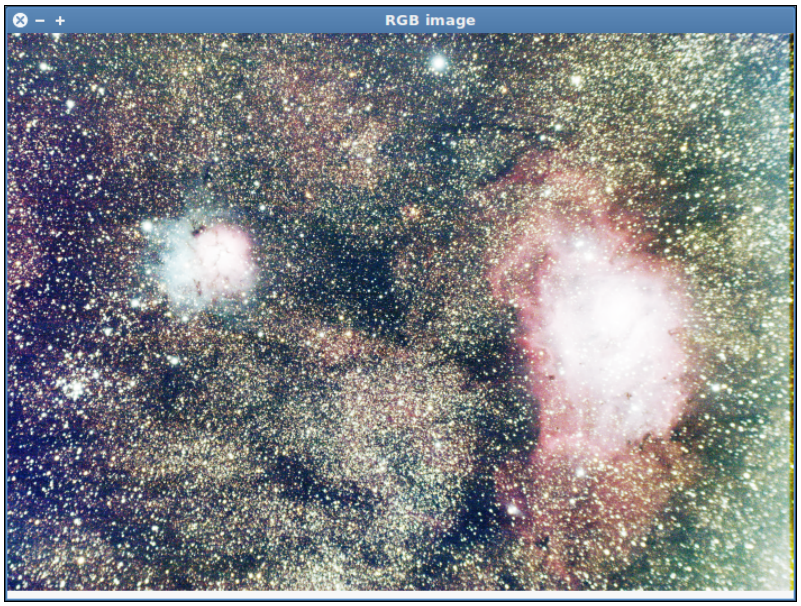
So, this step consists to remove this sky gradient. See the operation process below

Once you have a background model with no signal from nebula (use the display mode to check everything), you can apply the correction.



Two types of corrections have been implemented in Siril:

- Subtraction
- Division



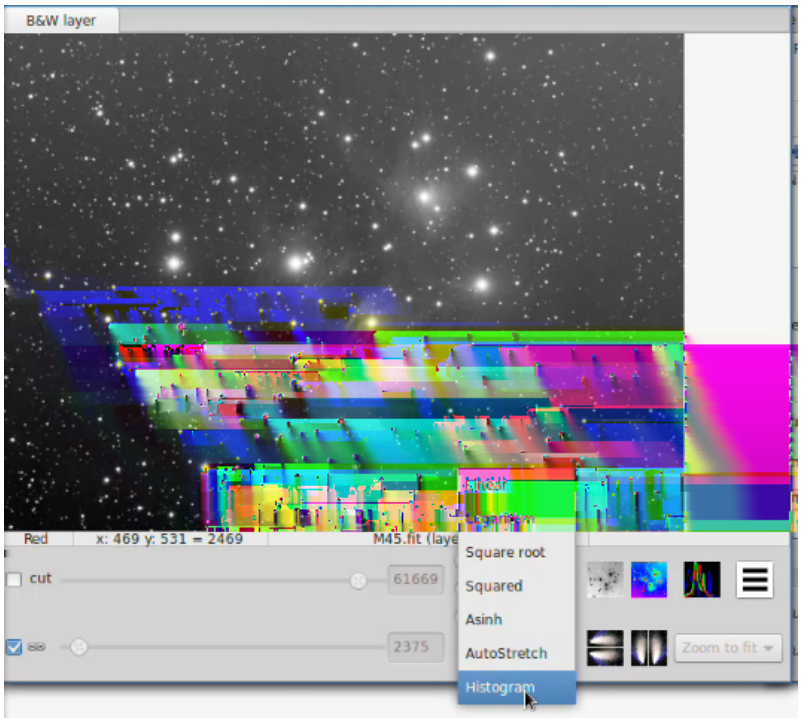
Additive phenomena like light pollution gradients should be corrected by subtraction but multiplicative effects like vignetting should be fixed by a division. But in the last case, the correct procedure is to correct the image with flat field during pre-processing step.

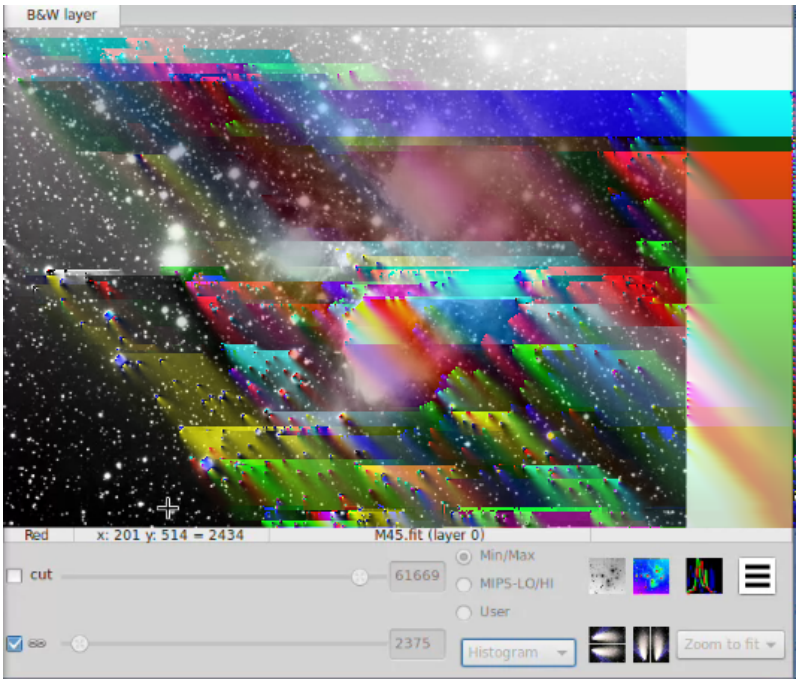
After the tool were applied, you should check the background neutralization, and, maybe, apply a new midtone correction.

Step by step background extraction

1 Image Display

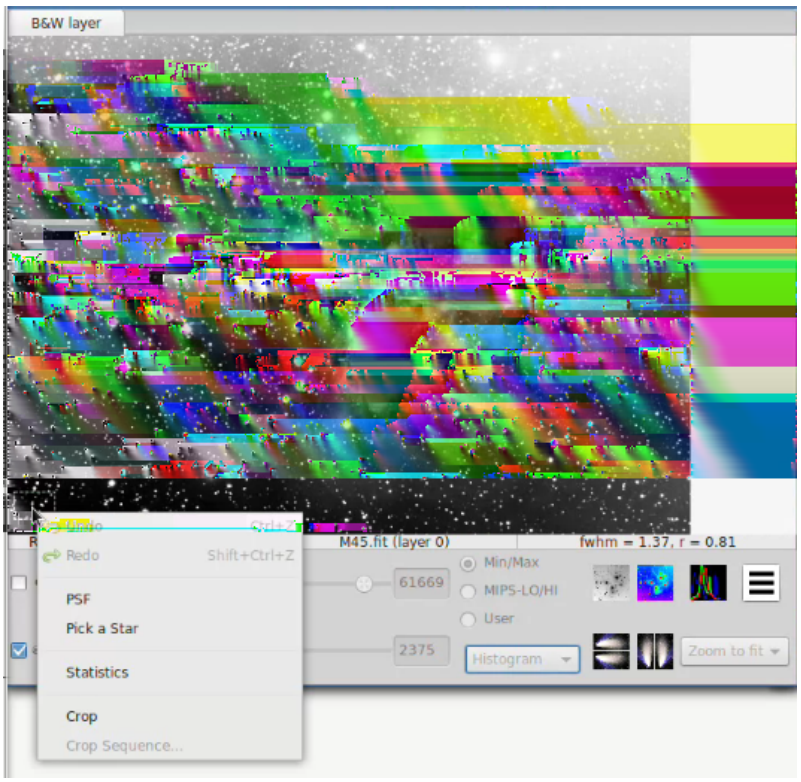
From the preview screen [p.73] choose the histogram[p.190] mode

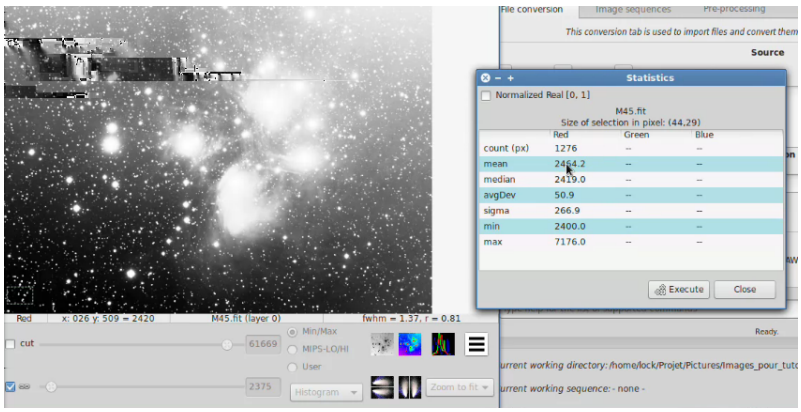




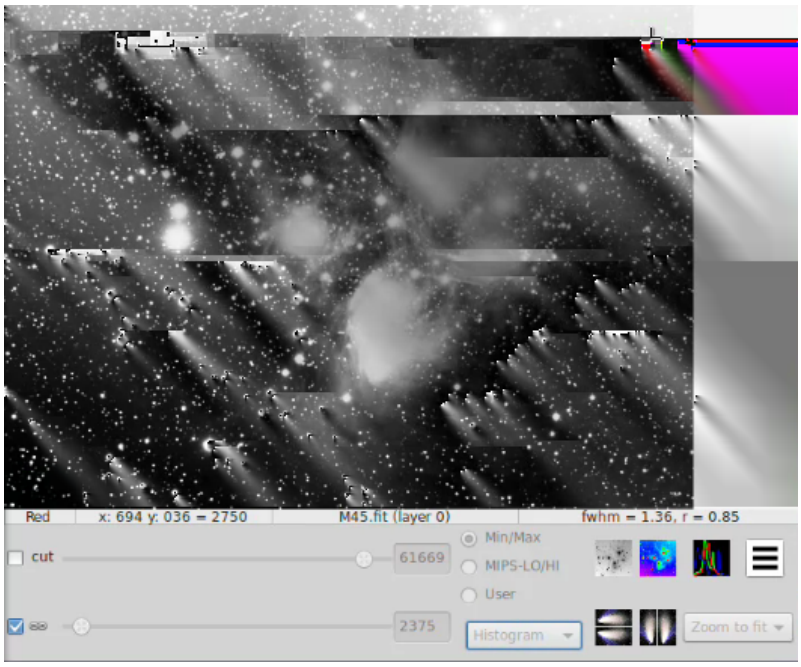
2 Show statistics

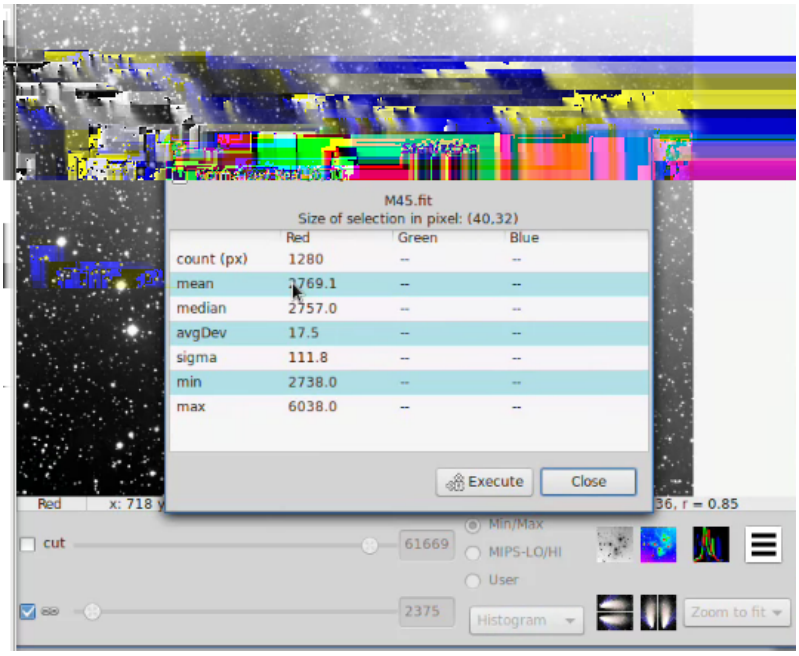
Select an image area





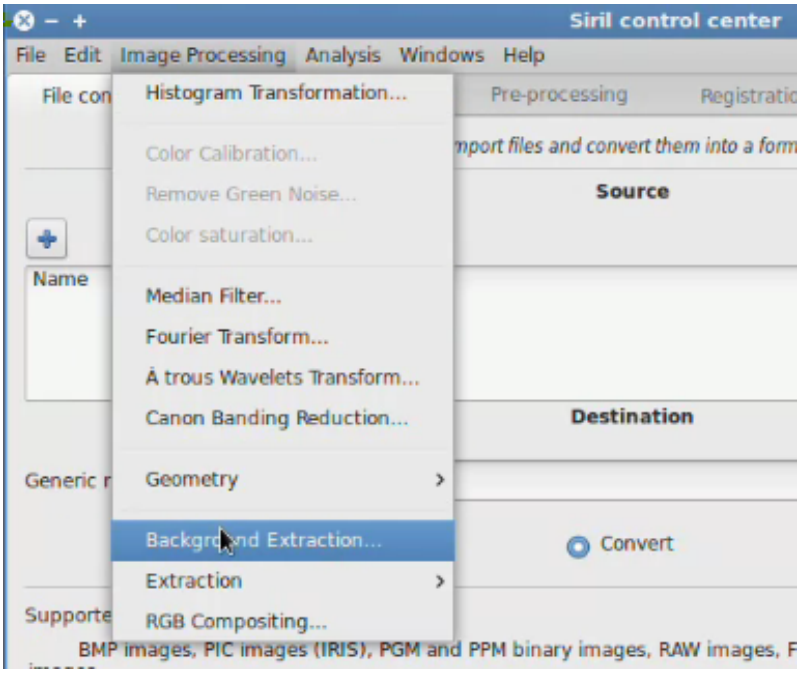
3 Select a second area

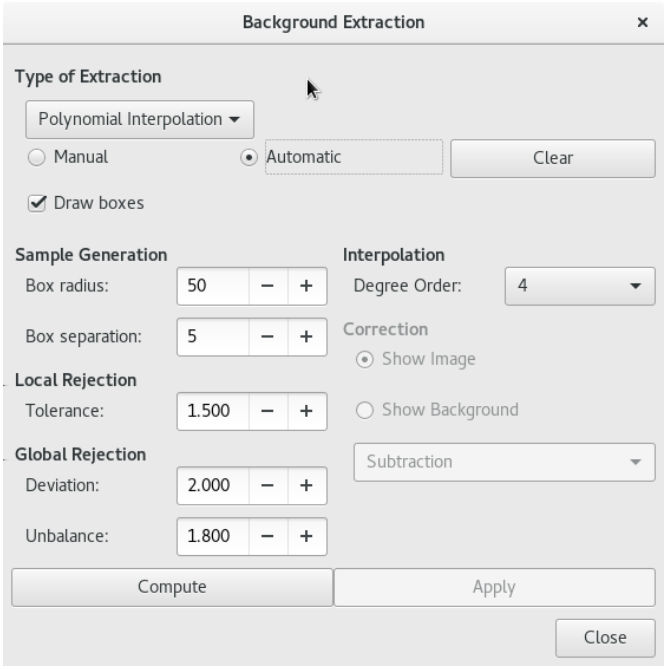




4. Values
5. Background extraction tool

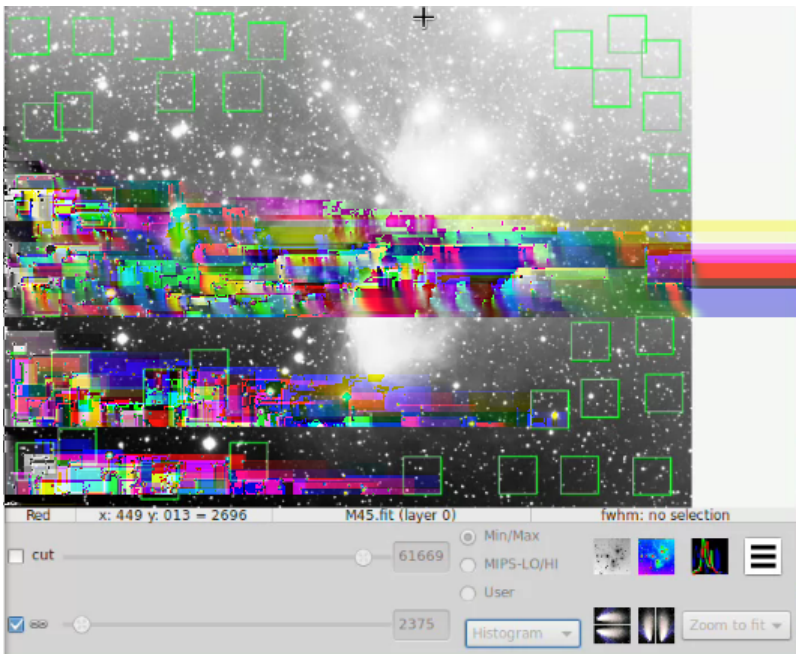
Menu ☰Image Processing/Background Extraction





6. Select ⦿Manual
7. Select Areas

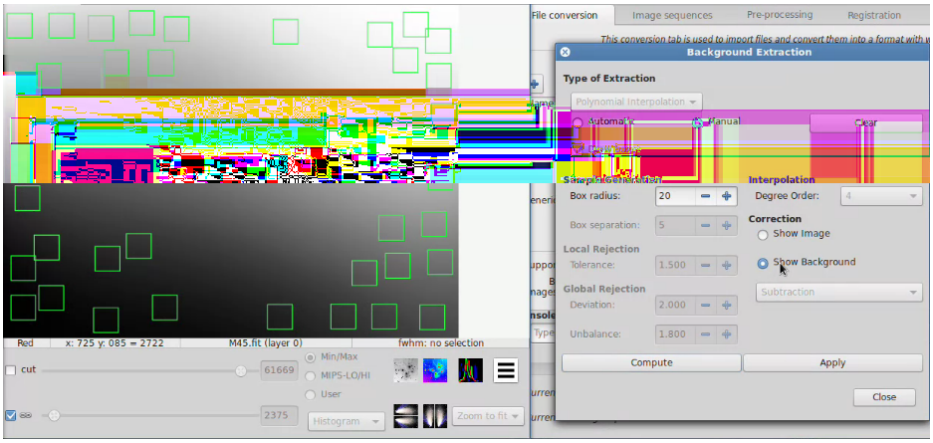
> You must select what appears to you to be the sky background.



8 Extraction

Click on Compute

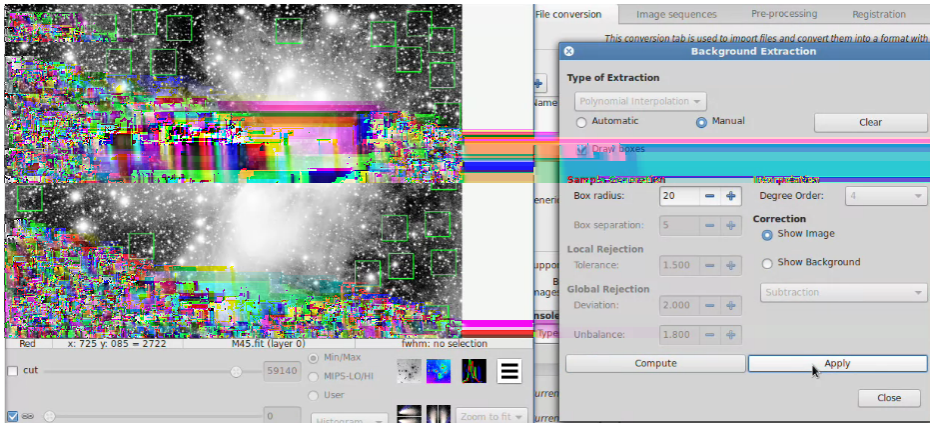
then check ⦿Show Background to display the result



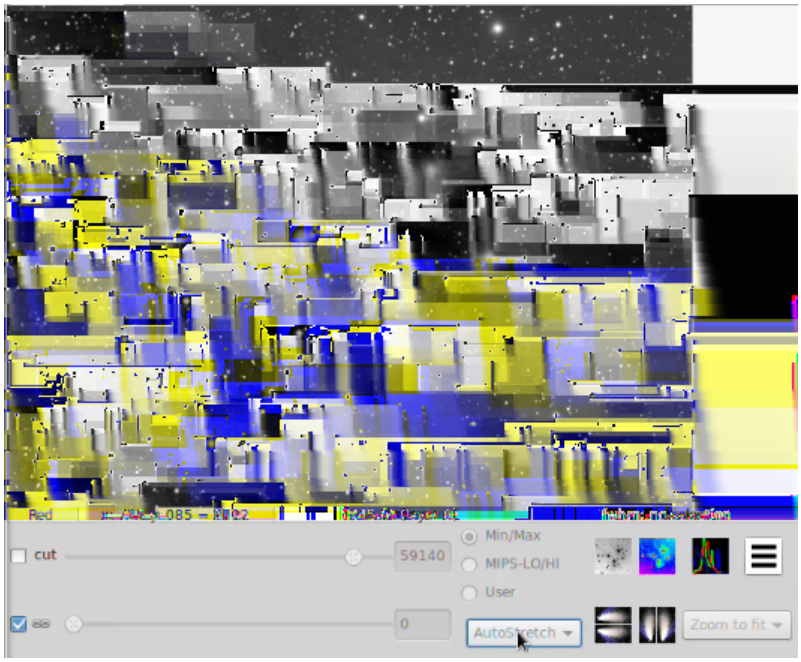
Result

Finish treatment

If you're satisfied by the result check ⦿Show Image then click on Apply and Close



Final result with a ☰AutoStrech display

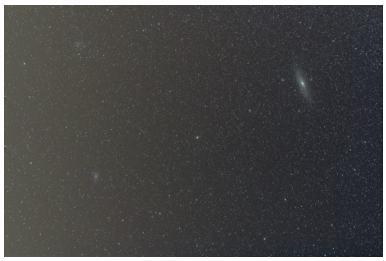


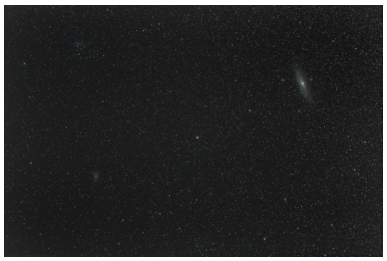
After extraction you can switch between image and background view in order to check the result. If you are satisfied you can thus apply the correction (subtraction or division). The 👉subtraction operation is very different from the simple operation used for the darks and bias/offsets.

Indeed, as the level of the background is identical to the one of the image, applying a simple subtraction would result in a image with a lot of black pixels. To avoid any inconvenience, the operation is done with 32-bit signed integers and finally converted into native format (16-bits unsigned).

Sample treatment

Following, respectively, the start image and final image (after background extraction with subtraction and equalization):





2.1.6.2. Color calibration tool

First, we need performing a balancing of the colors of a linear RGB. Siril provides a tool of color calibration.

Two steps are necessary:

- You must select a background area with no stars and click on "select" in the "black selection" frame. Click on "background neutralization" will neutralize the background by aligning median values.
- In the second step, you need to select the "white selection" area. So, make a selection containing a nebula part and click on apply. Be warn that black selection MUST be always selected. However, your image must be cropped before in order to remove any frame that could misleading the tool.

See the complete operation process below

Color calibration

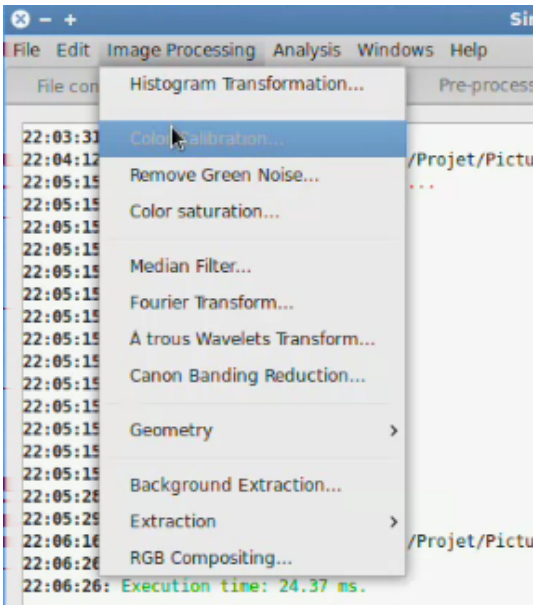
The colour calibration tool performs a balancing of the colours of a linear RGB deep-sky image.

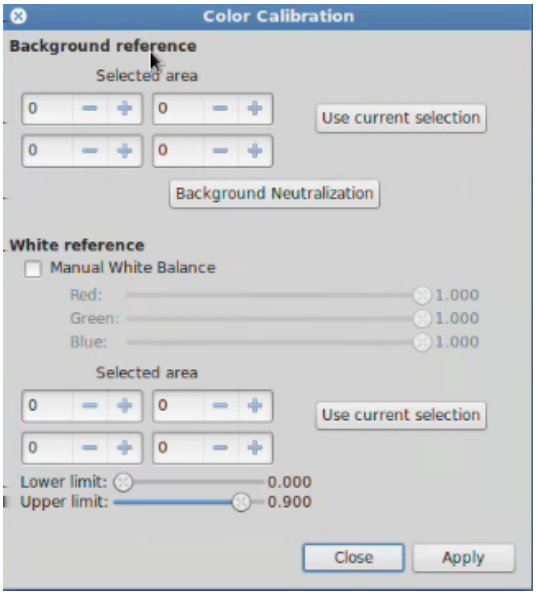
Note that Background Equalization (now called Background Neutralization) has been moved in the calibration box because it is needed before any calibration.

Prerequisite

You must have an image in memory

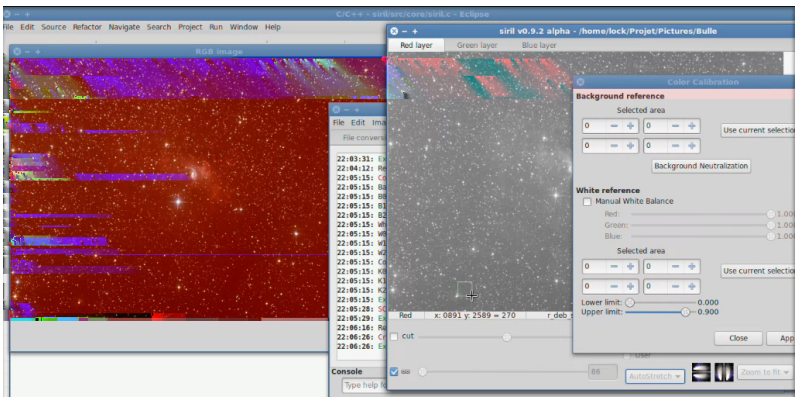
1. Open the Color Calibration menu





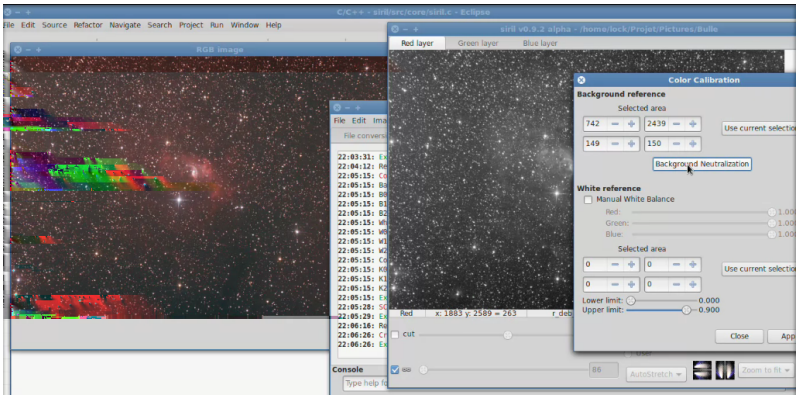
2 Neutralizing the background

Select a background area



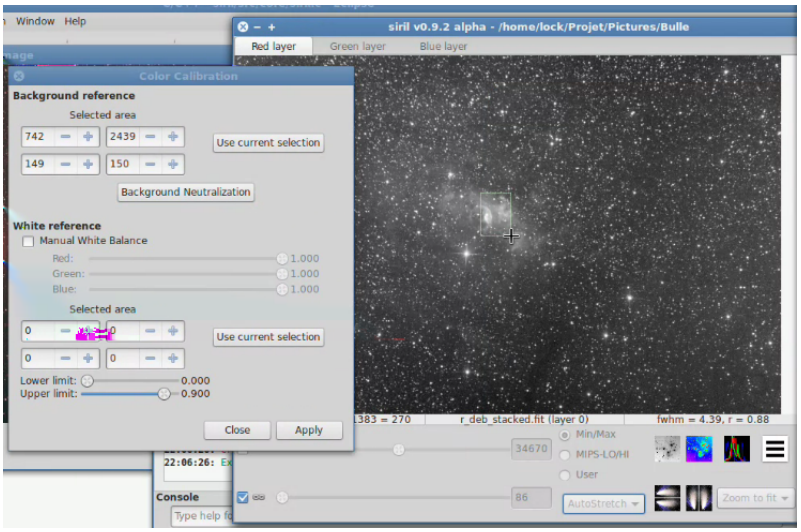
3 Click on Use current selection

4 Click on Background Neutralisation

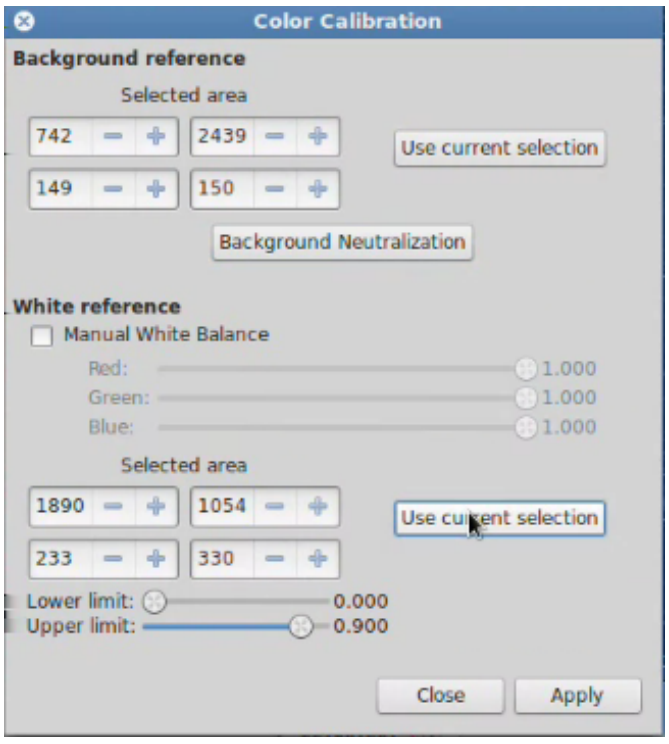


5 White reference

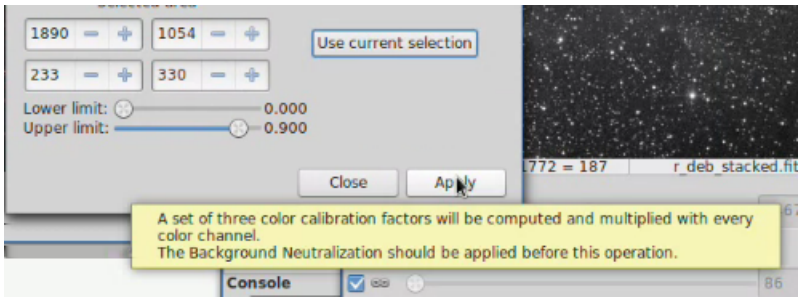
Select a reference area for white balance

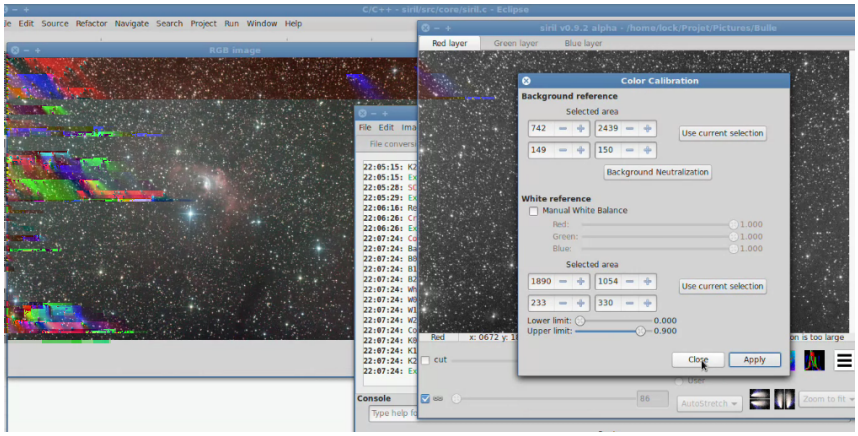


6 Click on Use current selection in the **White reference** zone



7 Click on Apply





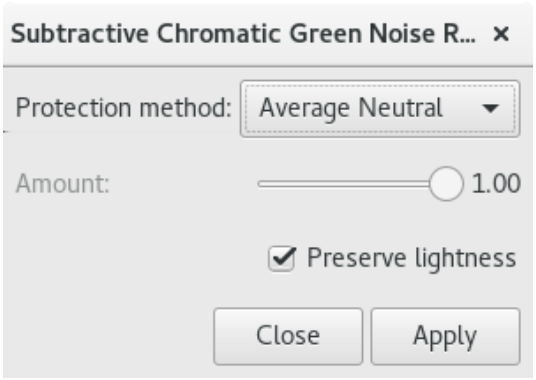
8 Click on Close to close the dialog box

9 **Reducing green tint in processed images**

This tutorial shows how to remove the green tint in processed images using the graphical user interface.

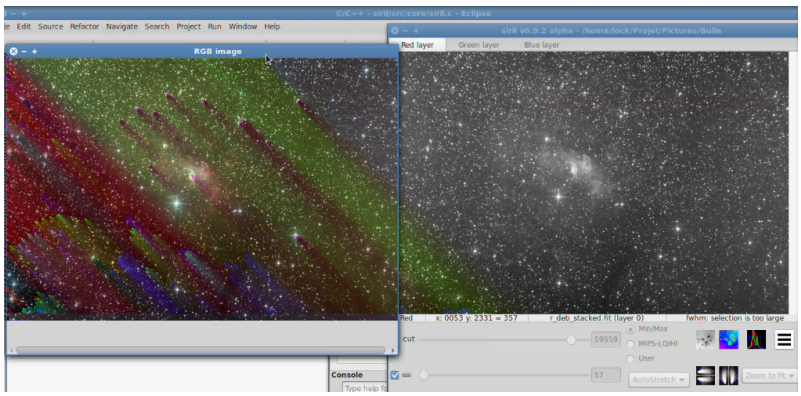
This function can also be accessed from the command line, with the *rmgreen* command.

Open the Image Processing/Remove Green Noise... menu



10 Click on Apply then Close

Result

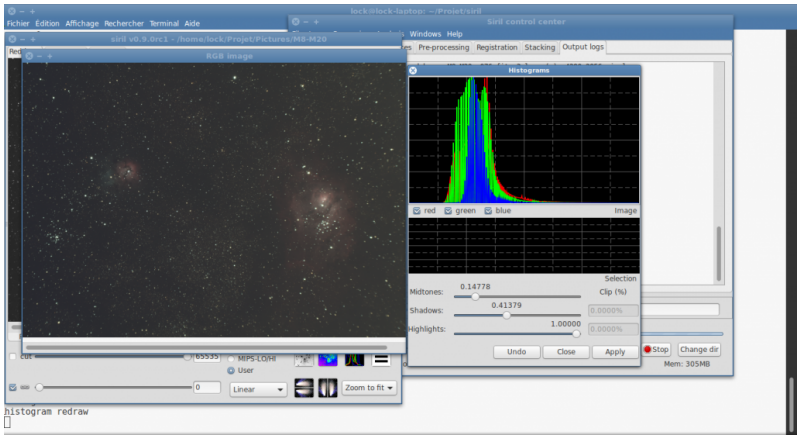


2.1.6.3. Green Noise Reduction

Last step in this processus is to reduce the green noise with the Remove Green Noise [p.89] tool.

2.1.6.4. Midtones Function Transfer

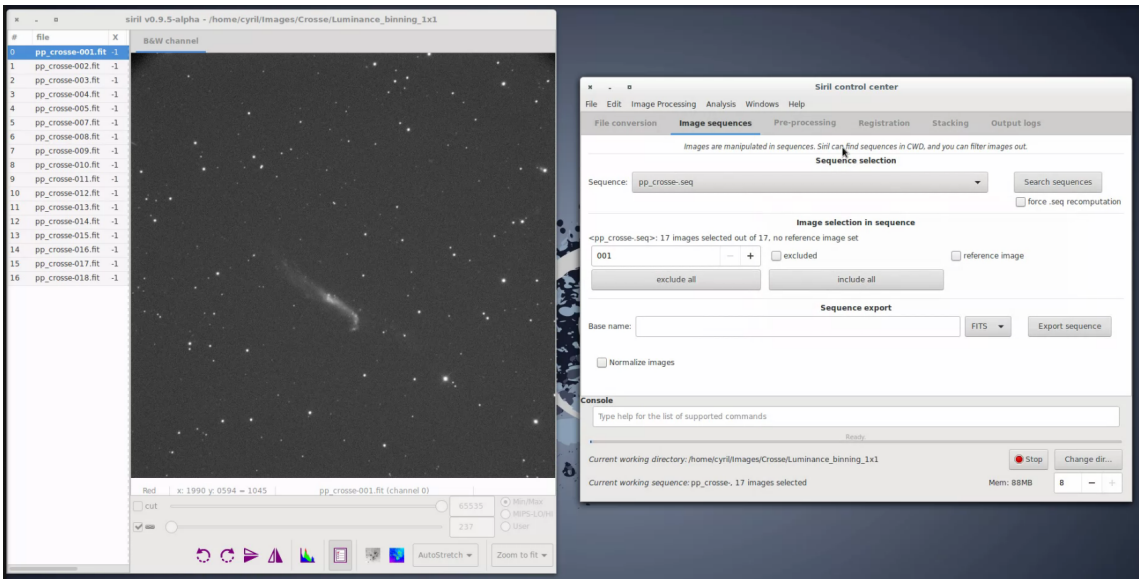
When you have a nice stacked and calibrated linear RGB, you can convert your image to a non-linear one. Siril provides an histogram transformation called Midtones Transfer Function also used in PixInsight[*http://pixinsight.com/doc/tools/HistogramTransformation/HistogramTransformation.html*] . By adjusting midtones sliders and shadows sliders iteratively (generally we do not use the highlight sliders) you will apply the histogram transformation. Pay attention to not clip to much pixels. Although Midtones Function Transfer is very similar to Gamma curves adjustment, they are not identical.



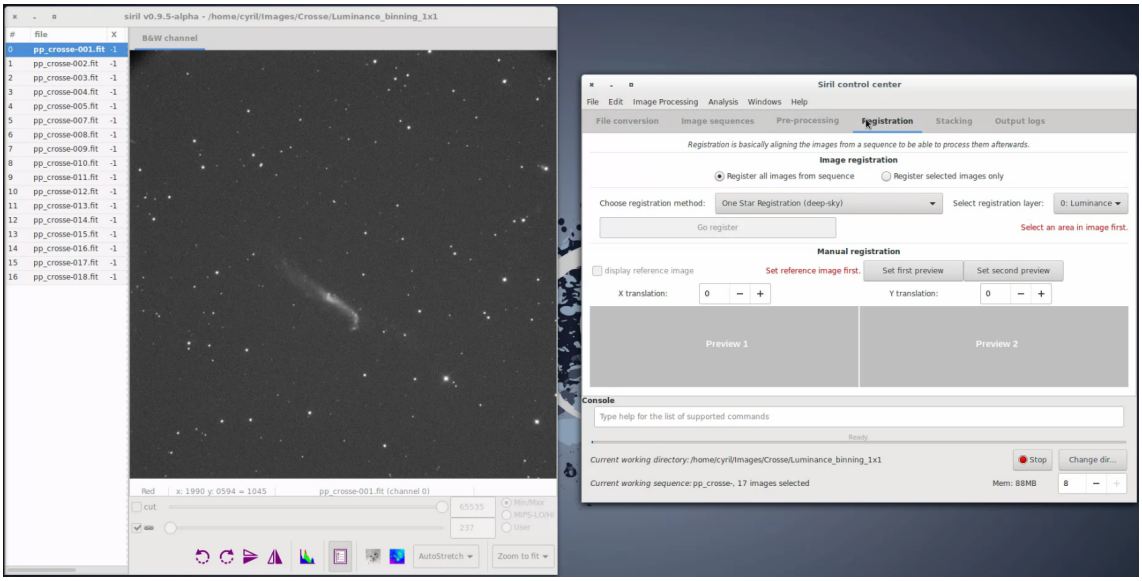
2.1.7. Manual Registration

In some case, manual is the best way to register sequences images.

1 Sequence selection

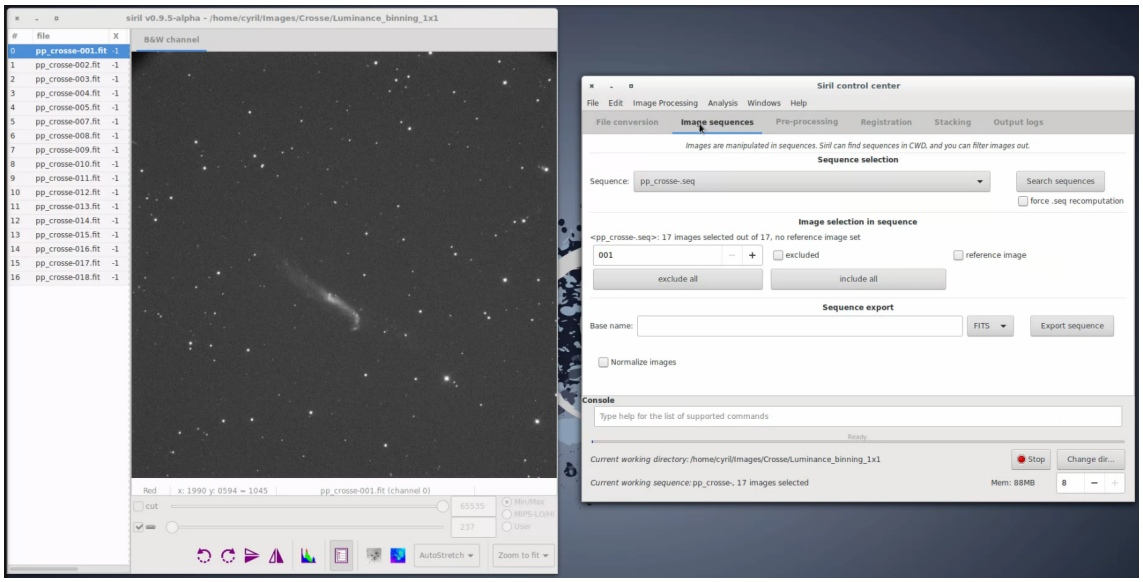


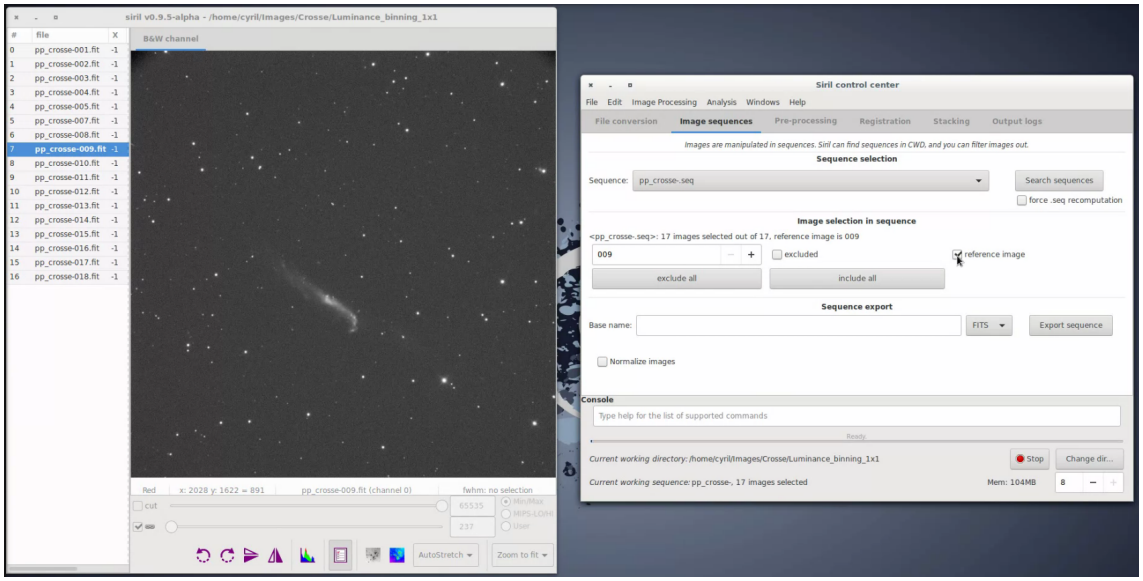
2 Registration



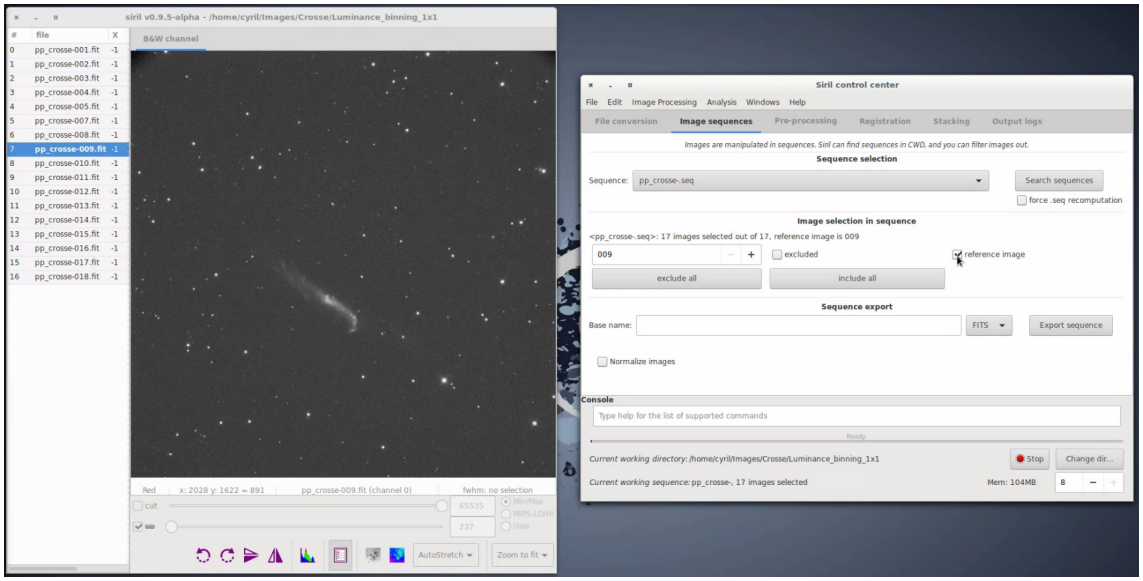
3 Image reference

Set image reference first

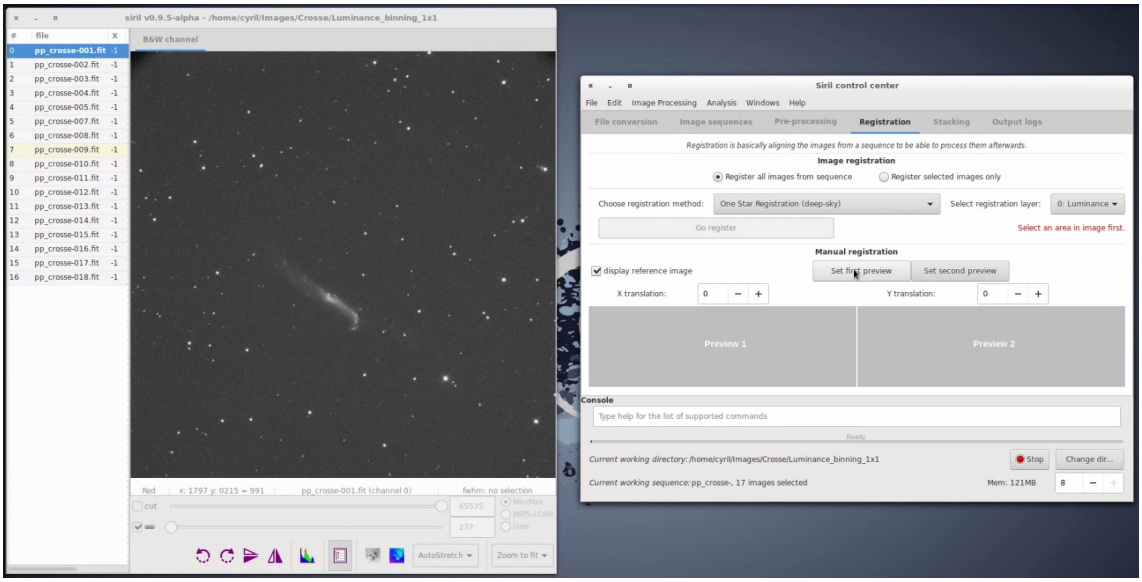


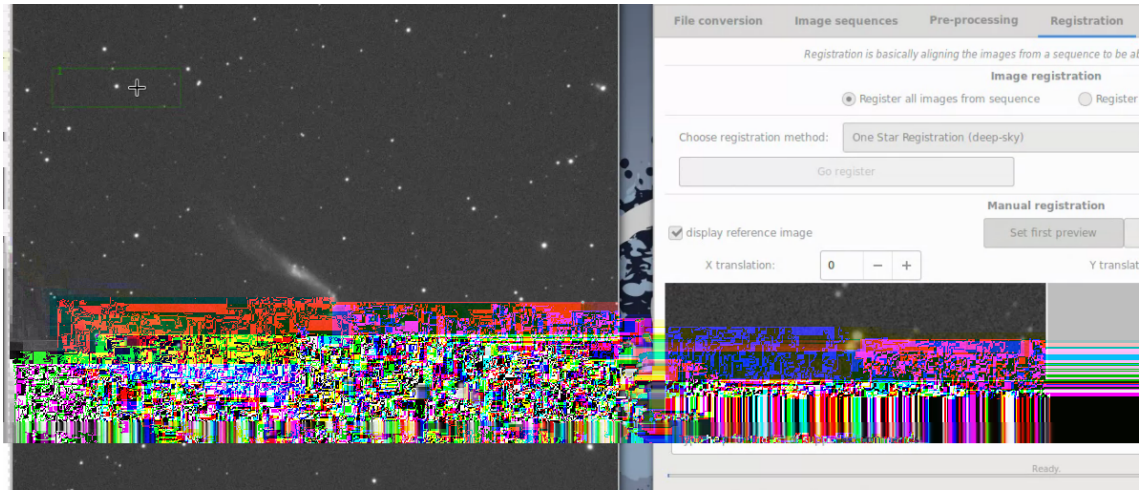


4 Register images

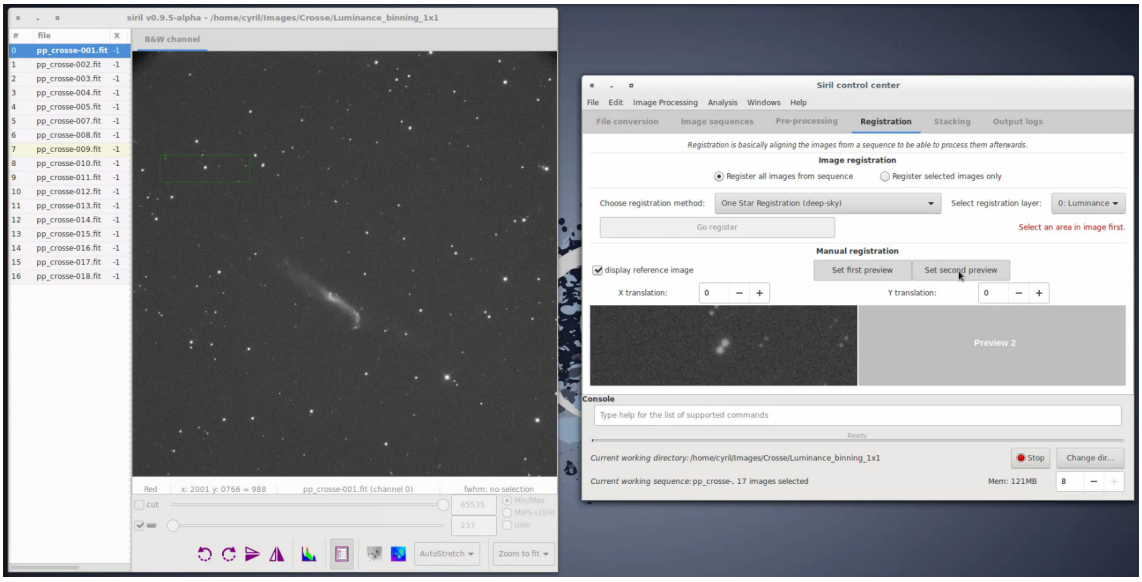


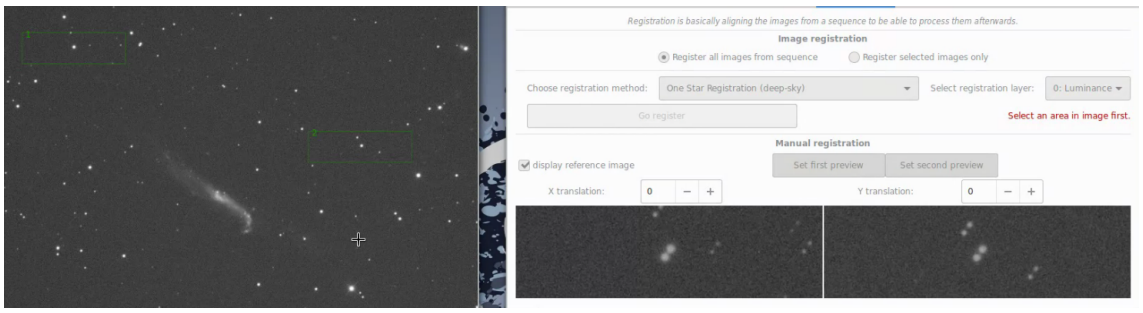
5 Set first preview



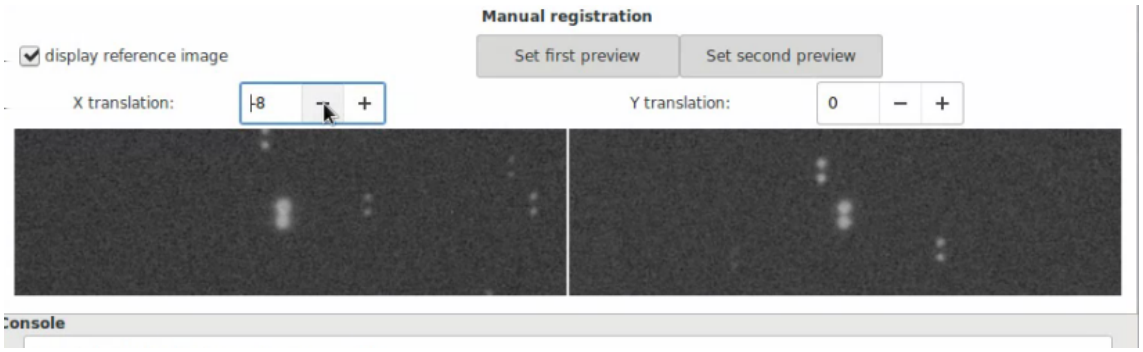


6 Set second preview

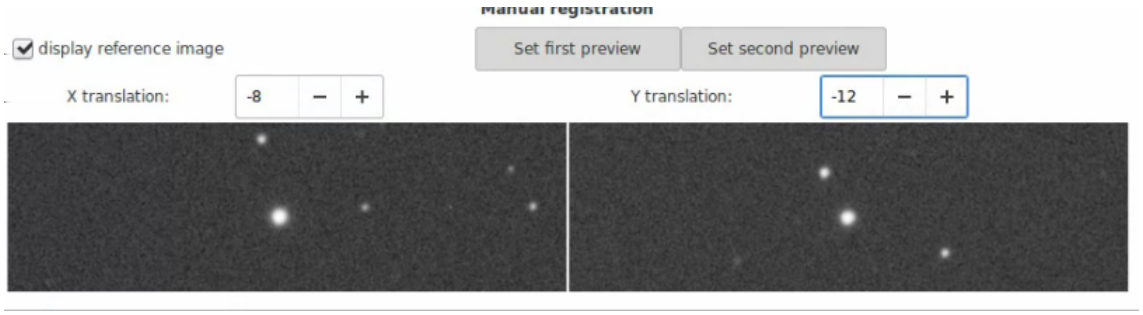




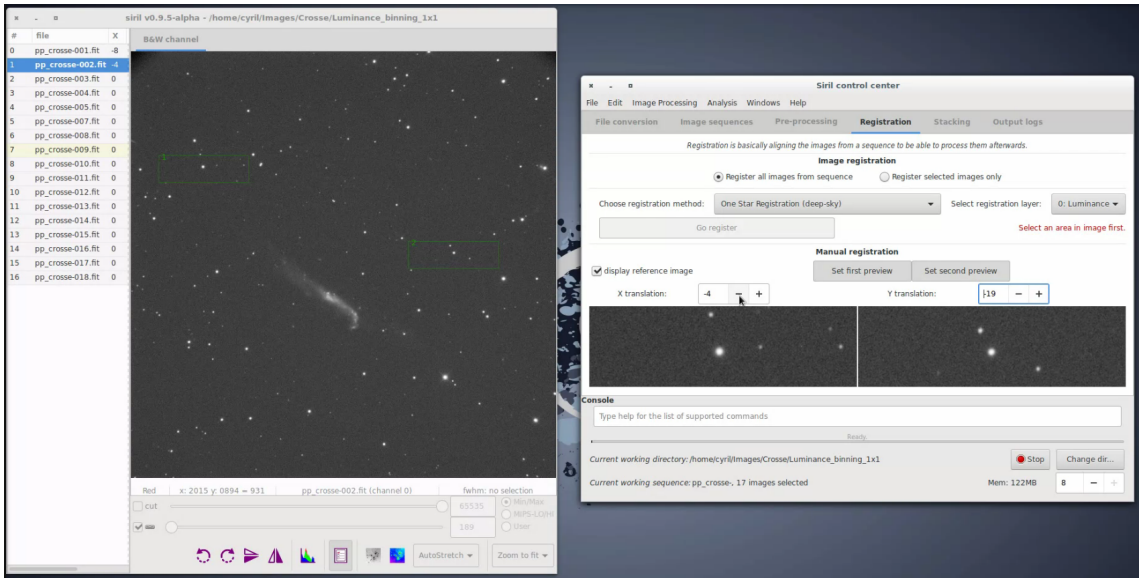
7 X translation

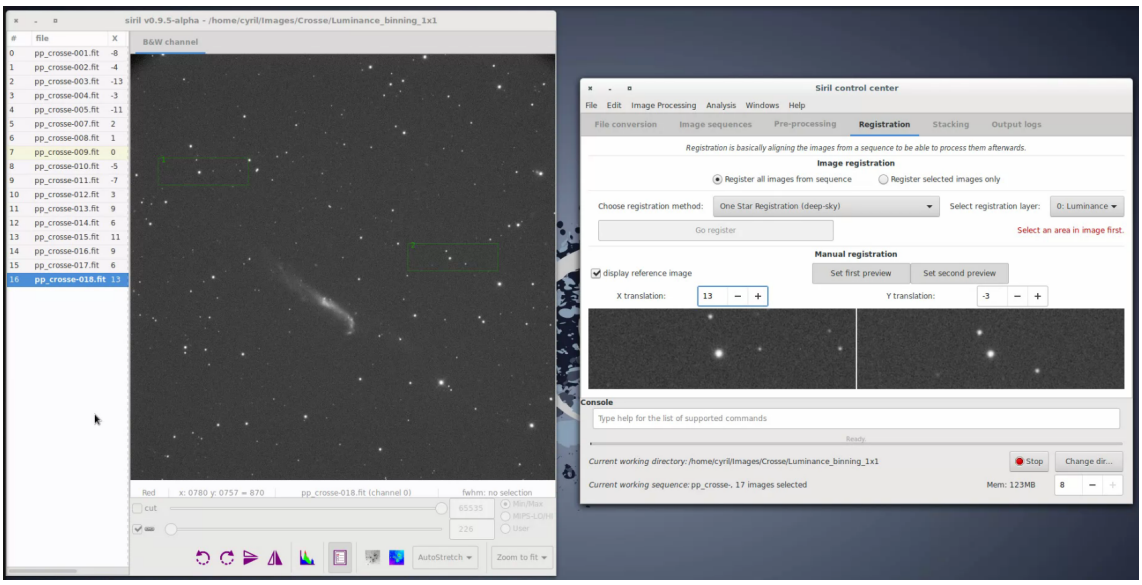


8 Y translation



9 Repeat operation for each images


siril v0.9.5-alpha - /home/cyril/Images/Crosse/Luminance_binning_1x1
B&W channel
Siril control center
Registration is basically aligning the images from a sequence to be able to process them afterwards.
Image registration
Register all images from sequence
Register selected images only
Choose registration method:
One Star Registration (deep-sky)
Select registration layer:
0: Luminance
Select an area in image first.
Manual registration
display reference image
Set first preview
Set second preview
Console
Type help for the list of supported commands
Current working directory: /home/cyril/Images/Crosse/Luminance_binning_1x1
Current working sequence: pp_crosse-, 17 images selected



10 Stacking

2.1.8. Aperture photometry

Prerequisite

Gnuplot is needed

Gnuplot[*http://gnuplot.sourceforge.net/*] is a portable command-line driven graphing utility for Linux

If you use Ubuntu or Mint click on the link below

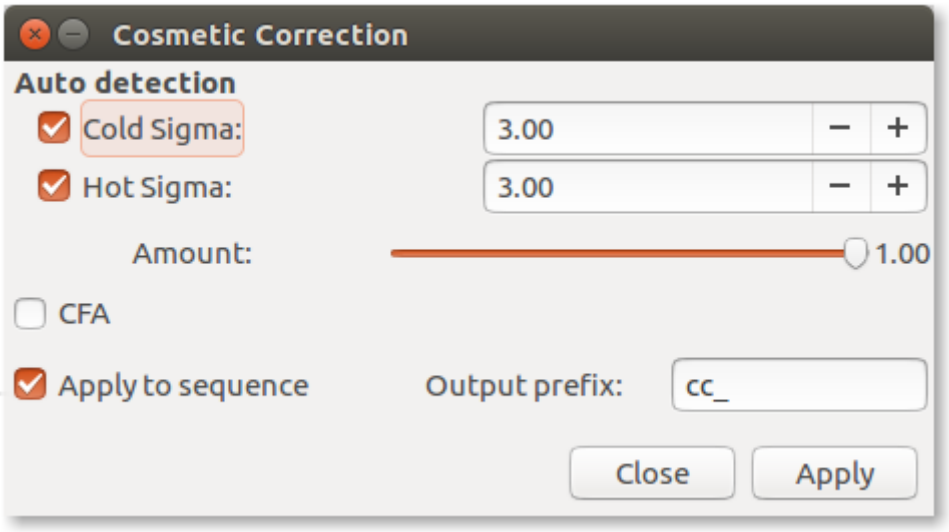
Gnuplot[*apt://gnuplot*]

1 Convert your images in a sequence

2 when preprocessed images

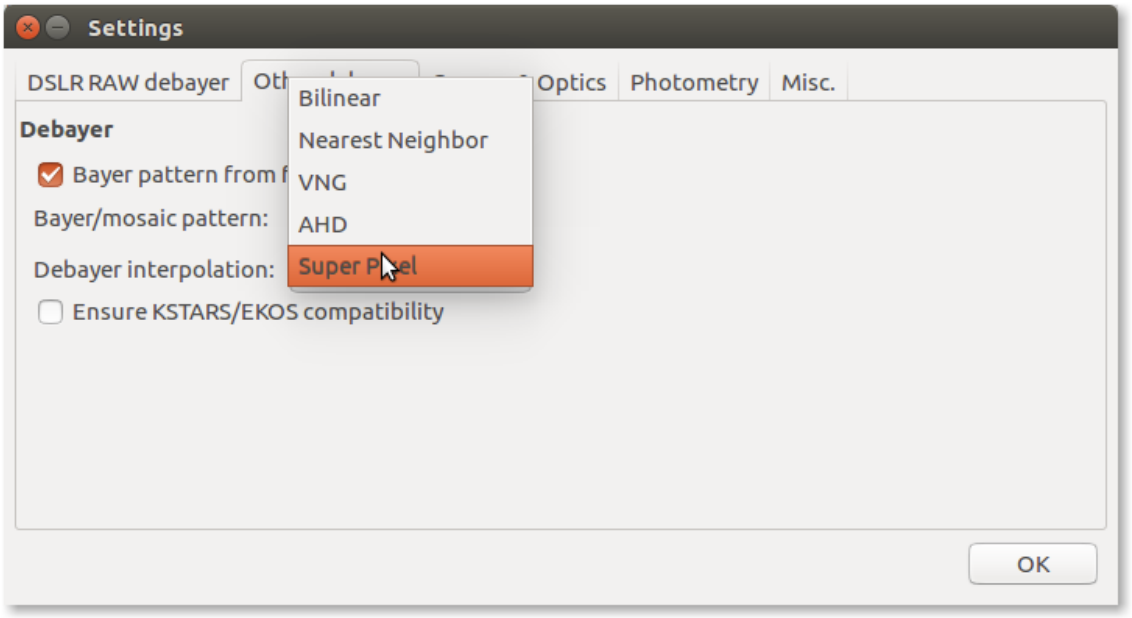
You can even consider making a cosmetic correction in addition to that proposed automatically with the the dark subtraction.

Menu ☰Image processing/Cosmetic Correction...



3. Demosaicing (for Bayer-Pattern images only)

★ **It is recommended to only work on the green channel, demosaiced with super pixel algorithm.**



⚠ Do not use other algorithms that would introduce artifact due to interpolation.

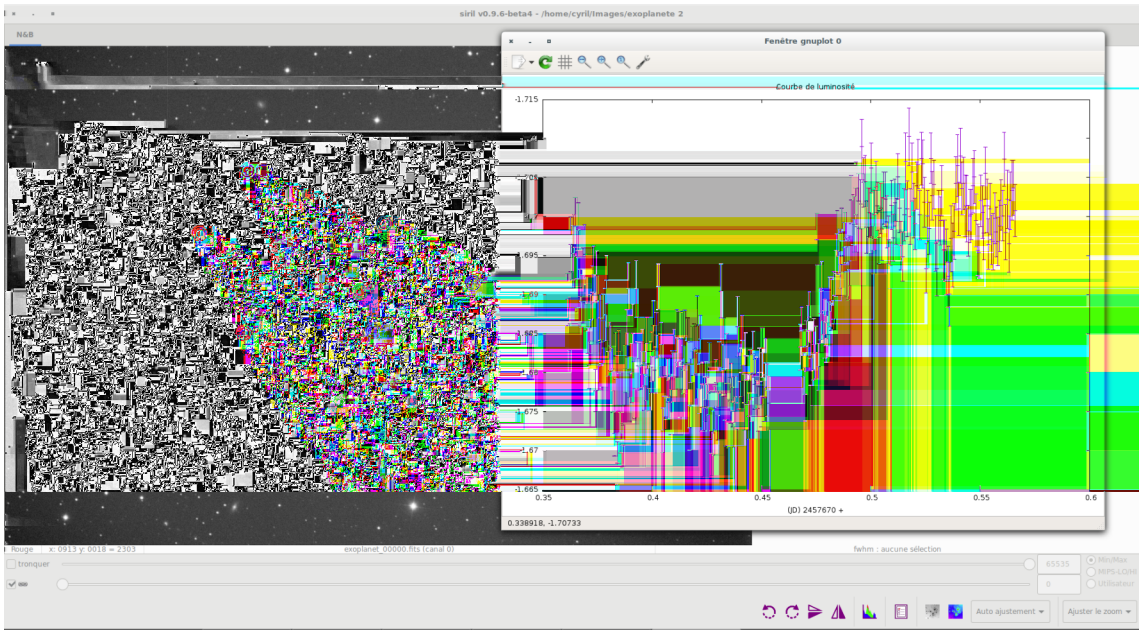
The red layer is strongly noised by the light pollution and the blue layer by atmospheric refraction.

★ Aperture photometry compares stars of the same image, keeping the green layer is the best choice.

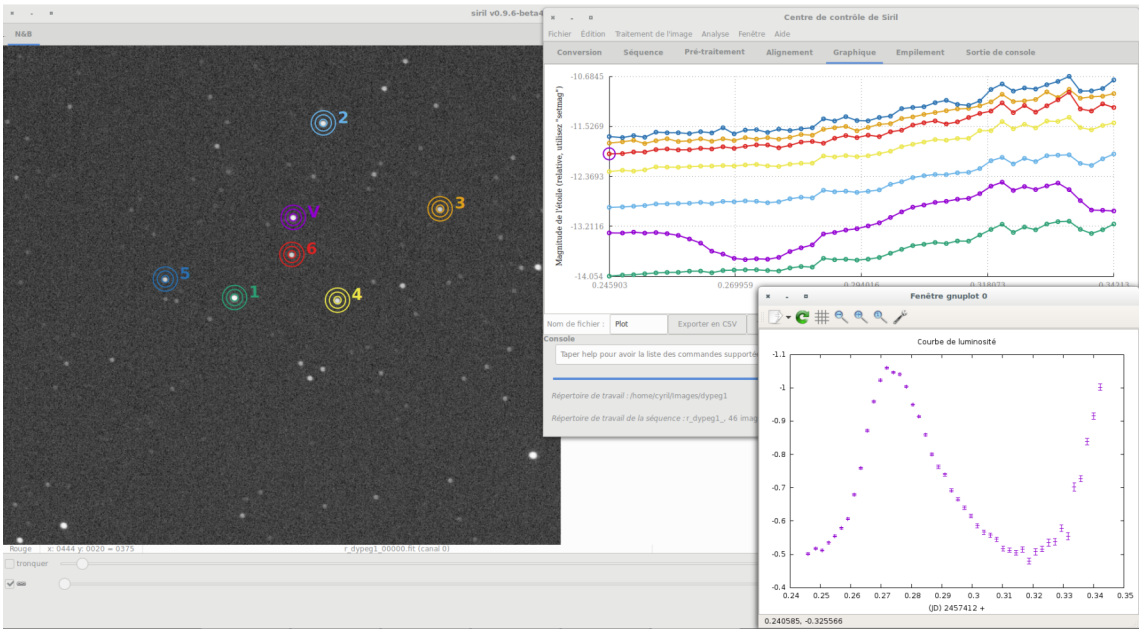
4. Images registration

Working on an registered images sequence is a better choice

Use either `One Star Registration` or `Global Star Alignment` in this case check ☑ Translation only



5. Export the sequence
6. Load the sequence
7. Select variable star



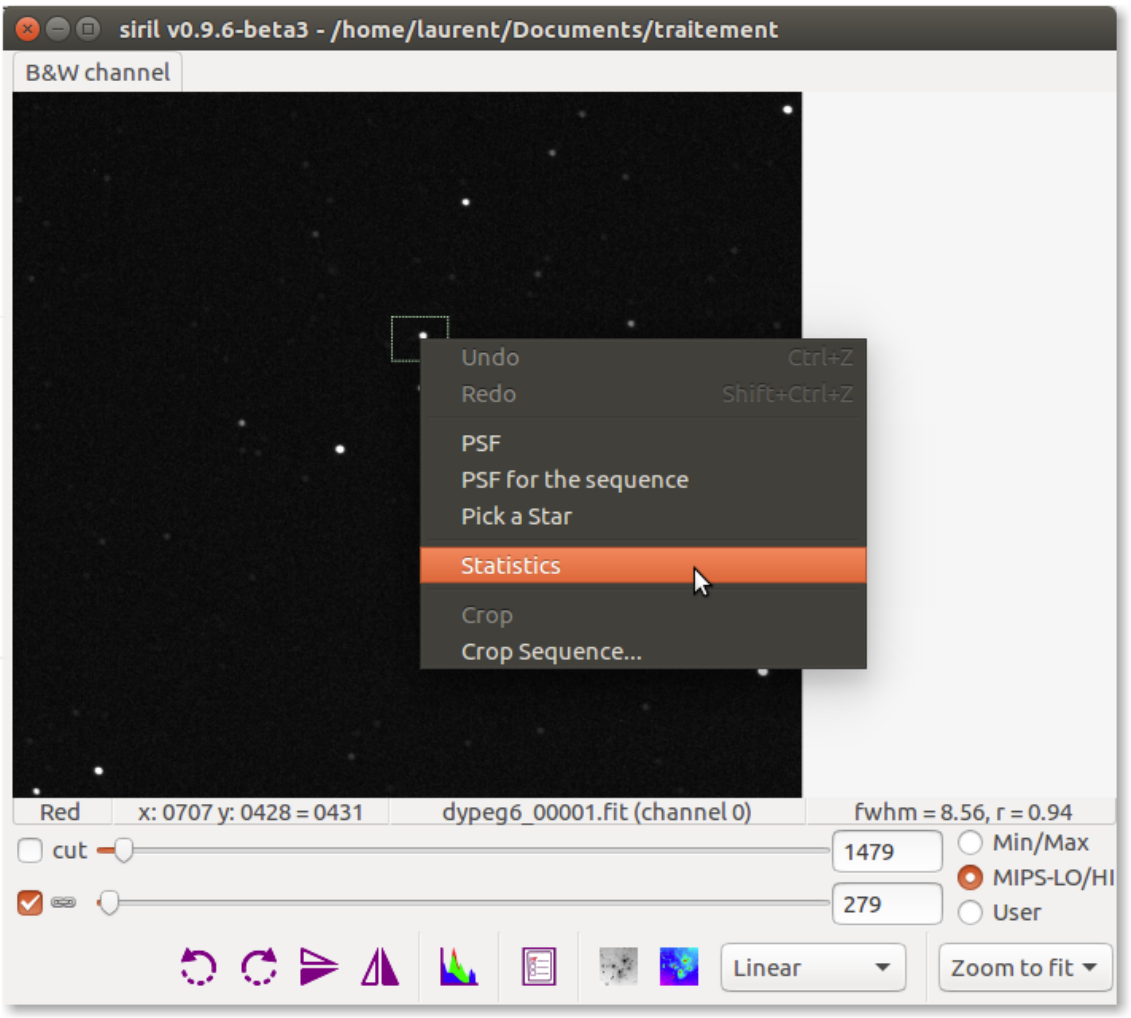
⚠ **You have to make a fairly wide selection,**

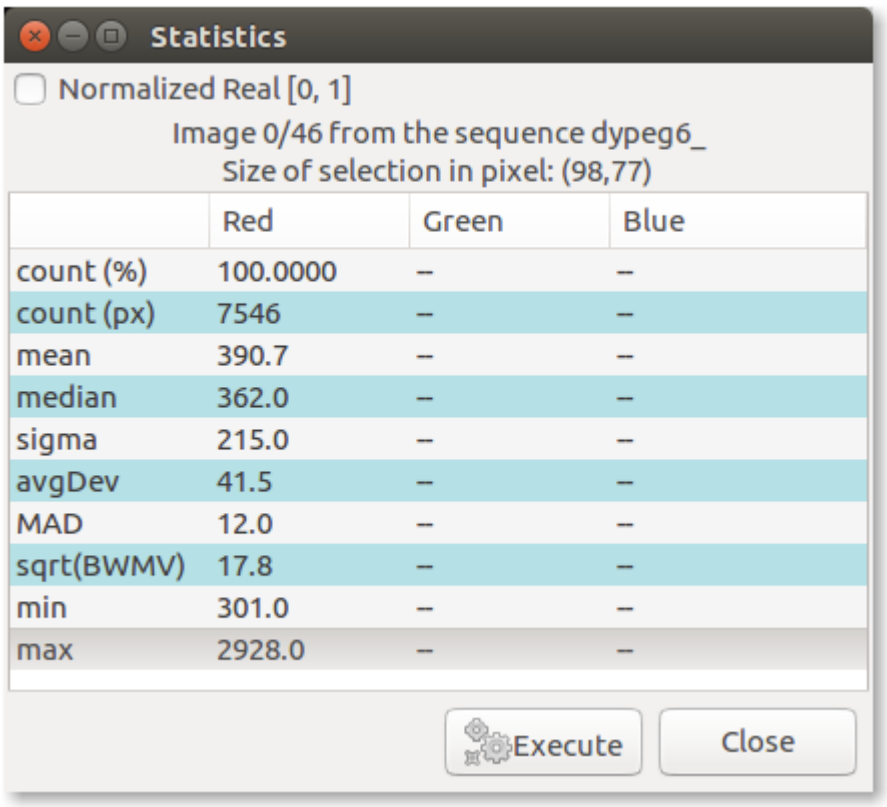
It is necessary to check that the largest circle is completely included.

All parts of the circle outside the selection will be ignored.

8. Once the selection has been made,

check with the statistics option that the maximum is not 65535

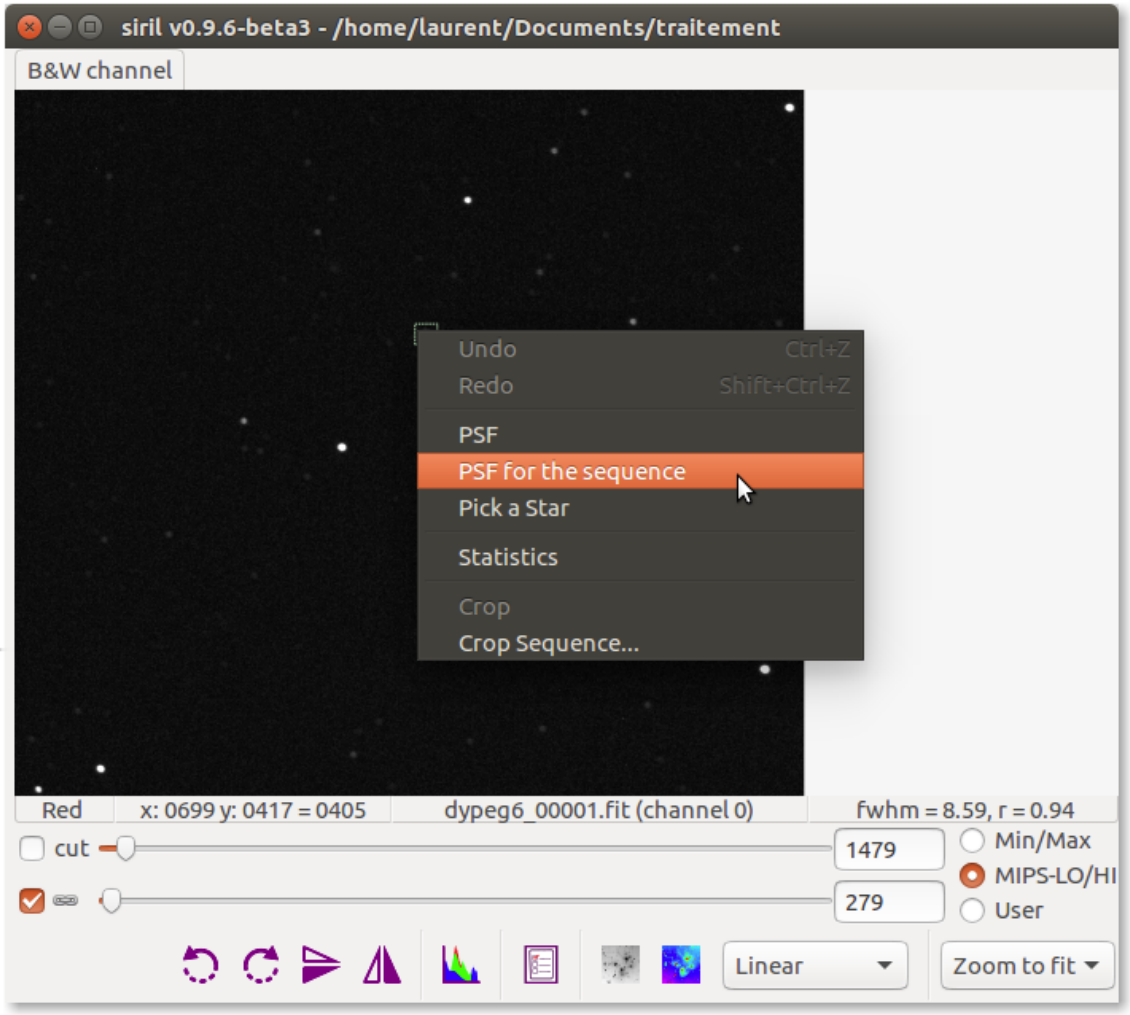




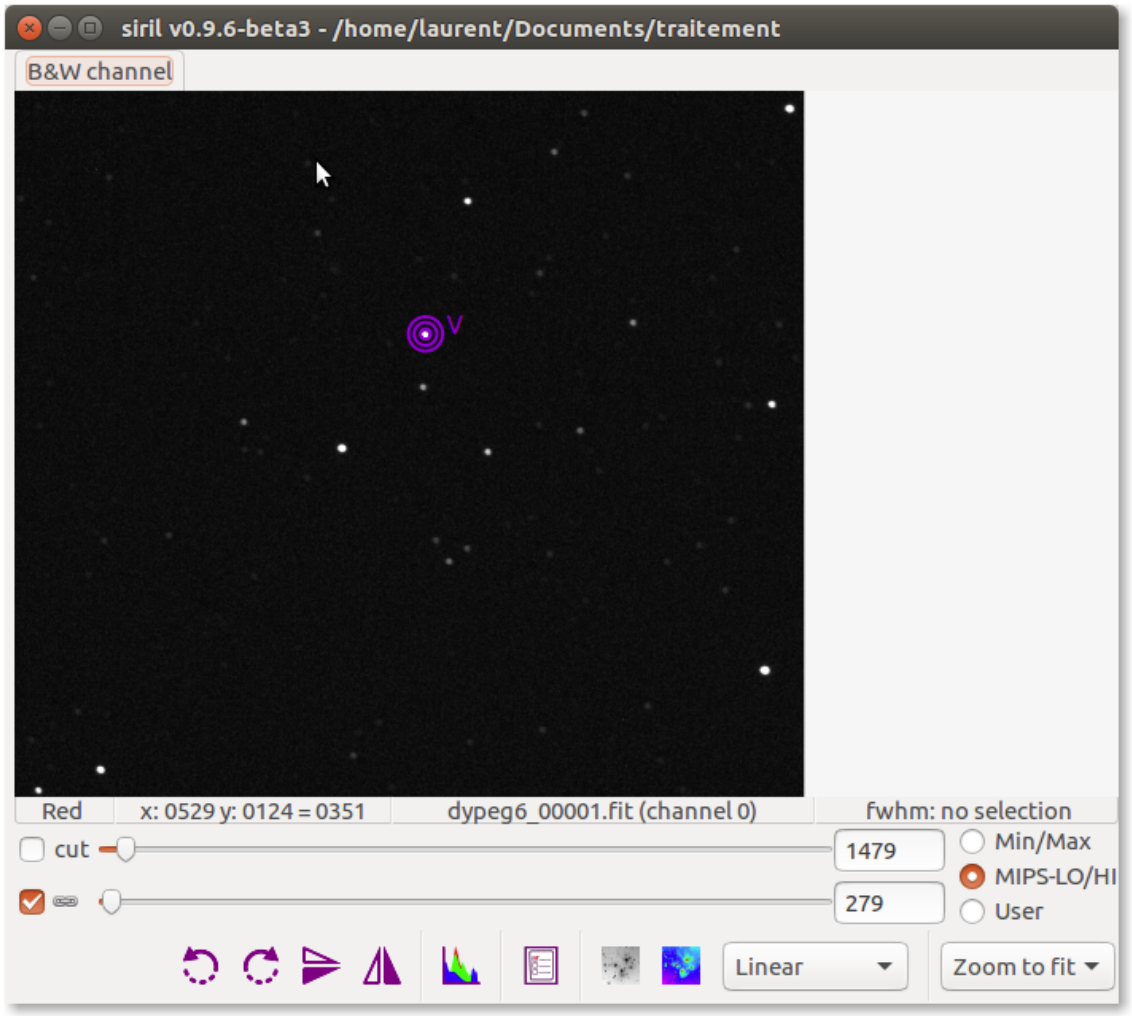
If the value is 65535, the star is saturated.

The results will be inoperable.

9 Do a *PSF for sequence*



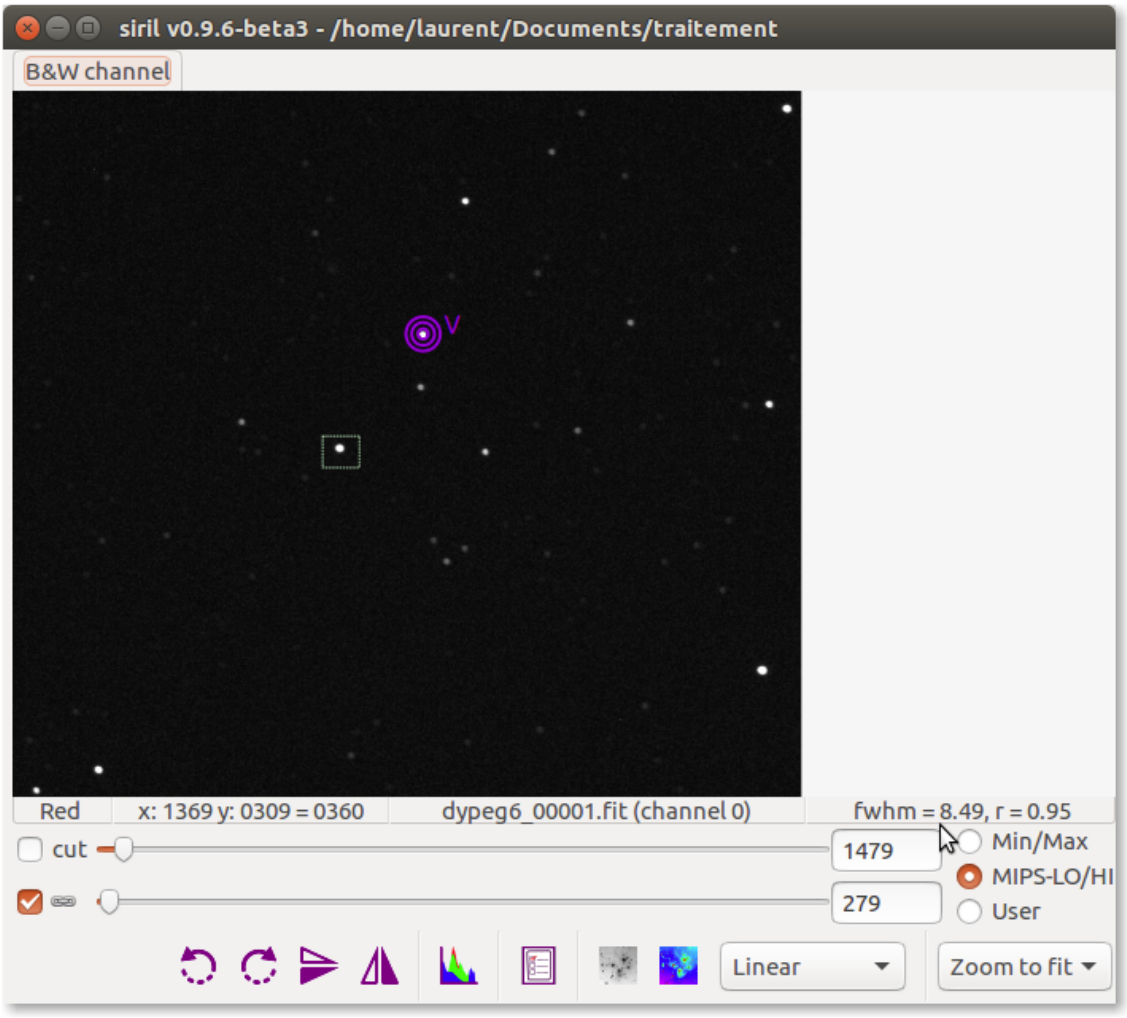
A median filter is applied to the selection in order to remove hot pixels that could lead to bad star detection.



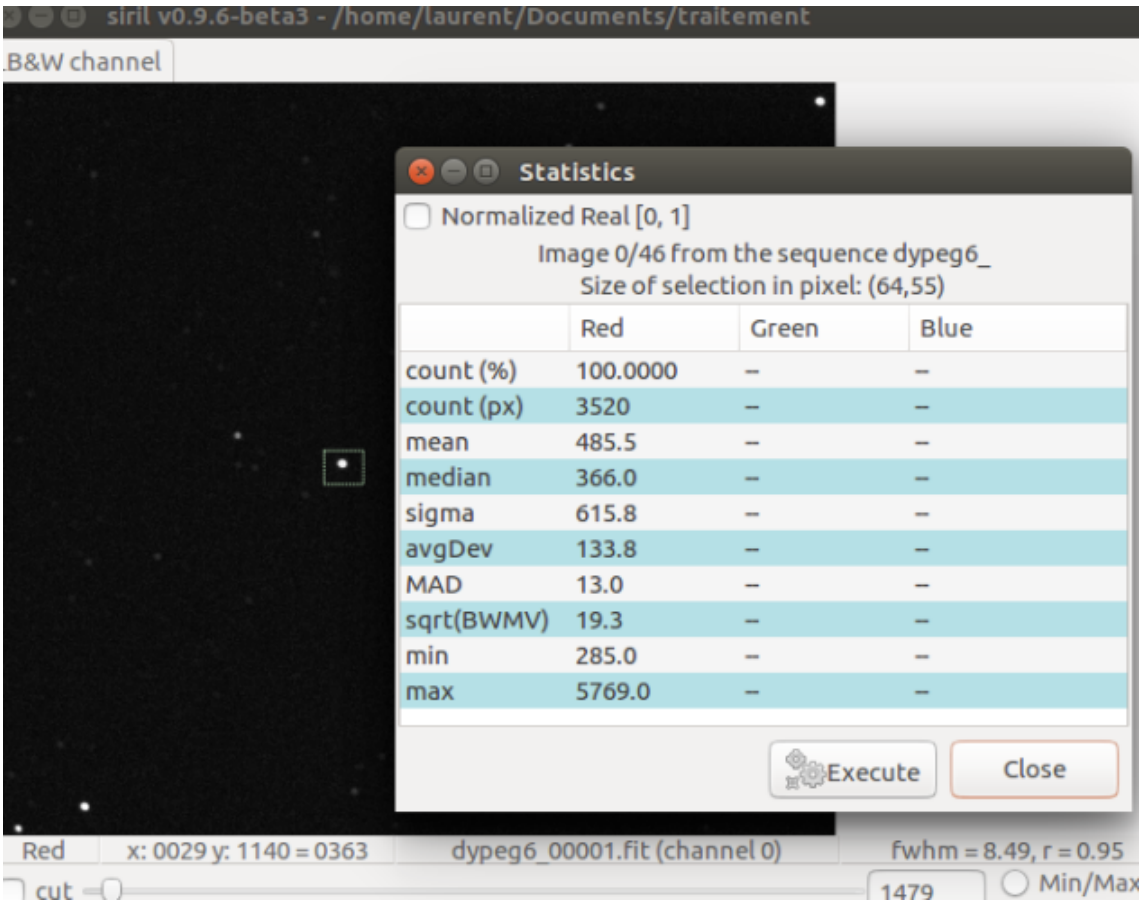
10 Then we look for stars with a close magnitude

To help you, when you make the selection, look at the bottom right of the image the value of the **FWHM** which must be close to the **FWHM** value of the variable star.

Here the value is 8.56 for the variable star.

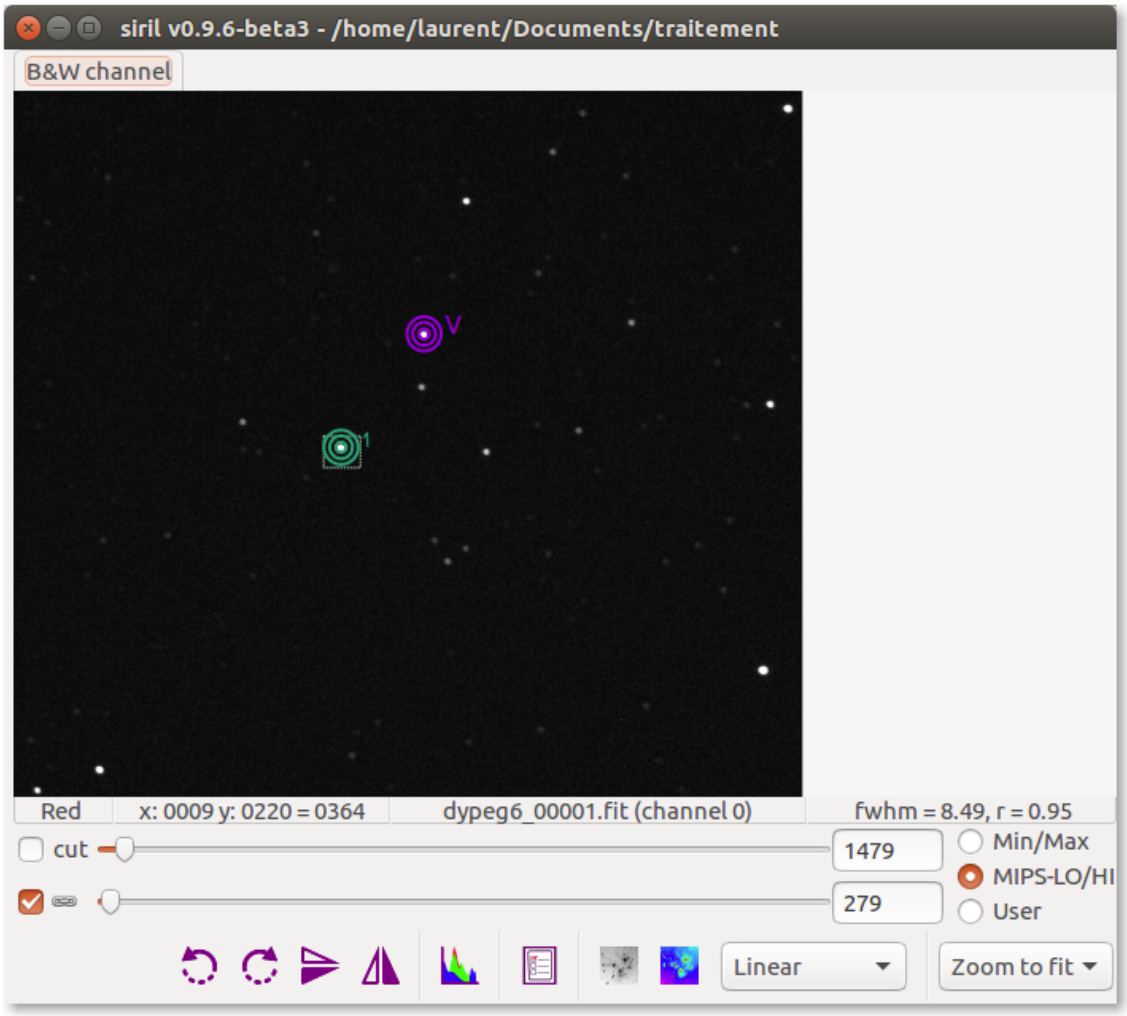


11 Check with Statistics tool that the selected star is not saturated



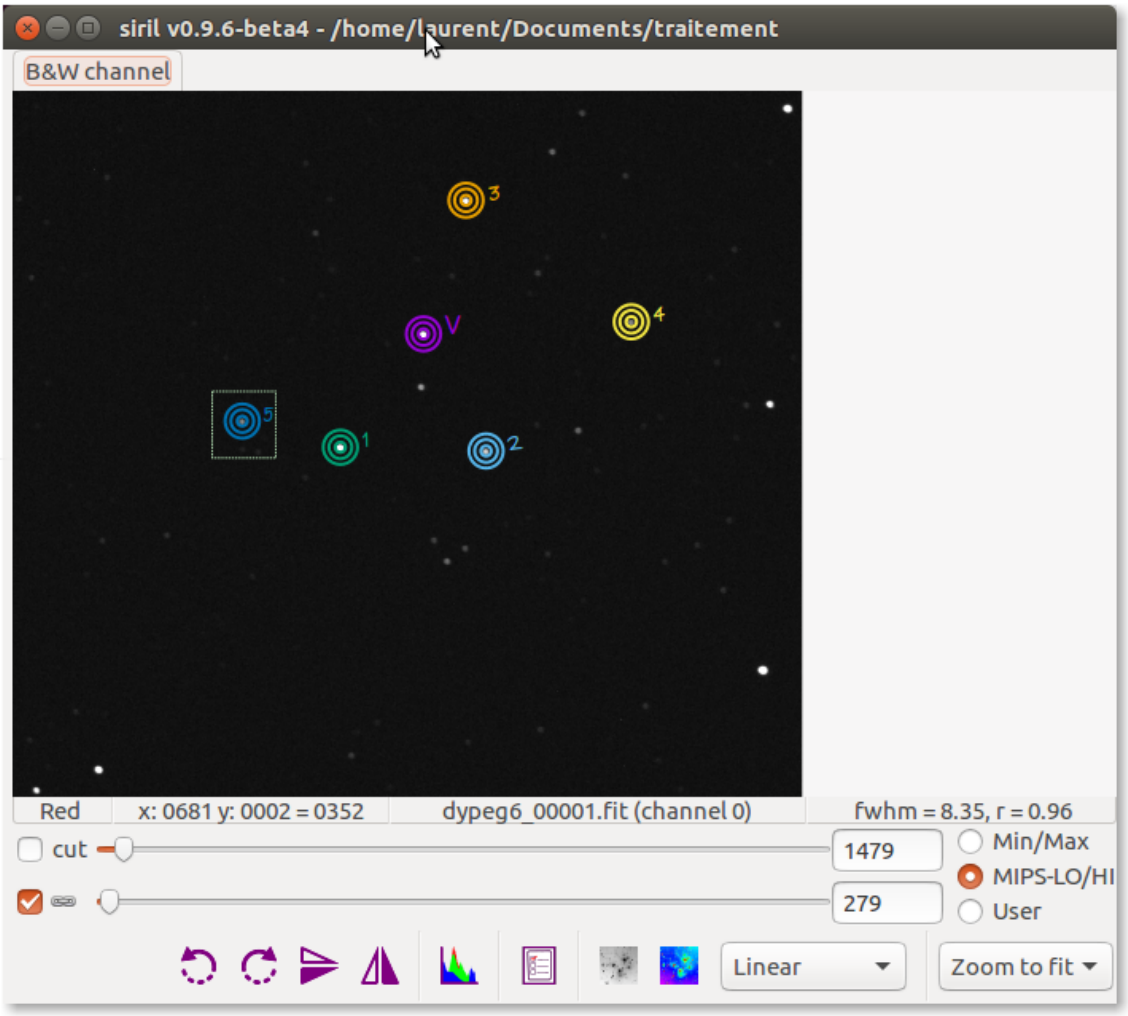
> The values given by the statistical tool only concern the open image, not the entire sequence of images.

12 Do a *PSF for sequence*

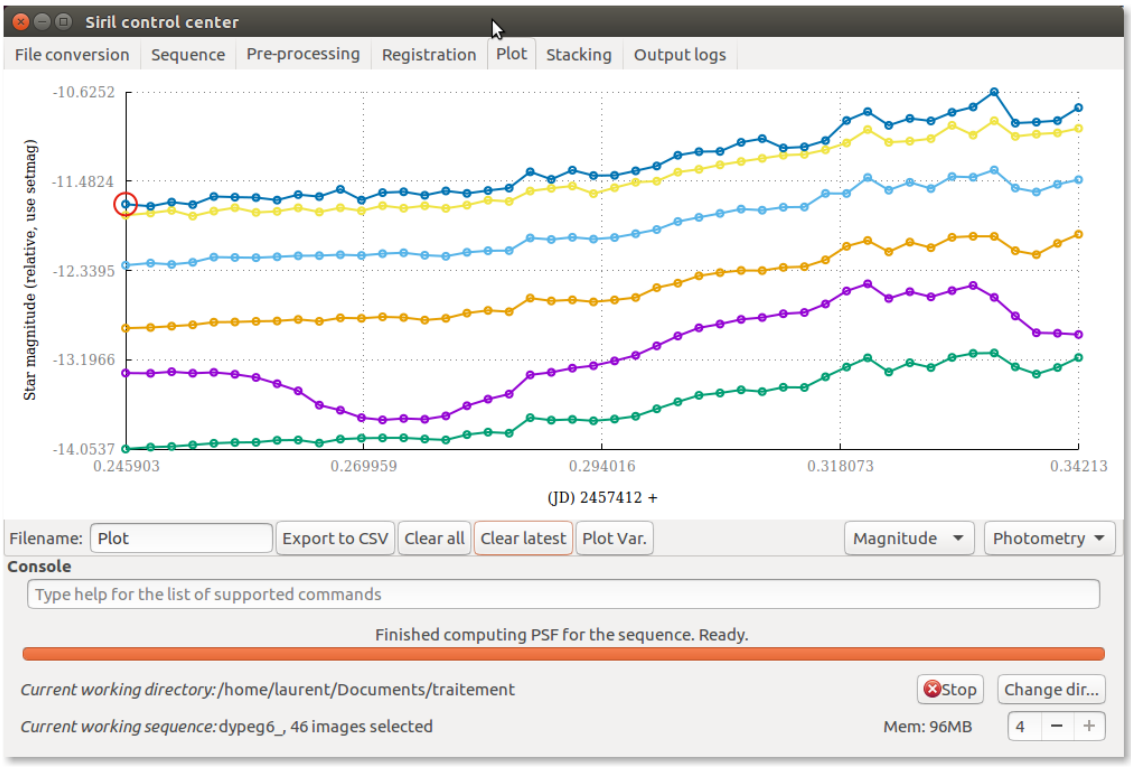


13 Repeat previous steps for 3 or 4 more stars

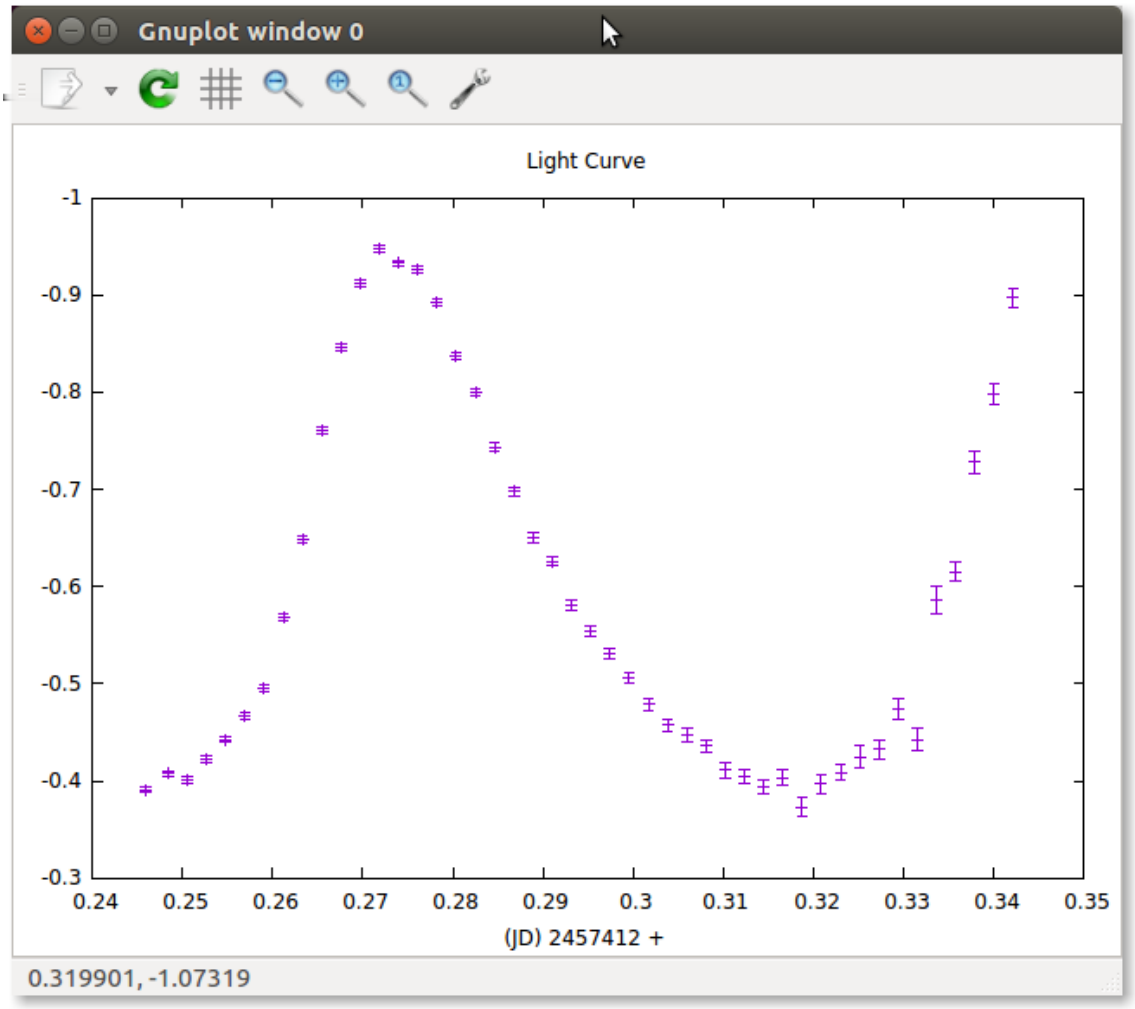
If a curve of a reference star appears suspicious, click on delete the last curve and choose another star.



14 In the plot tab select Magnitude



15 Click on Plot Var



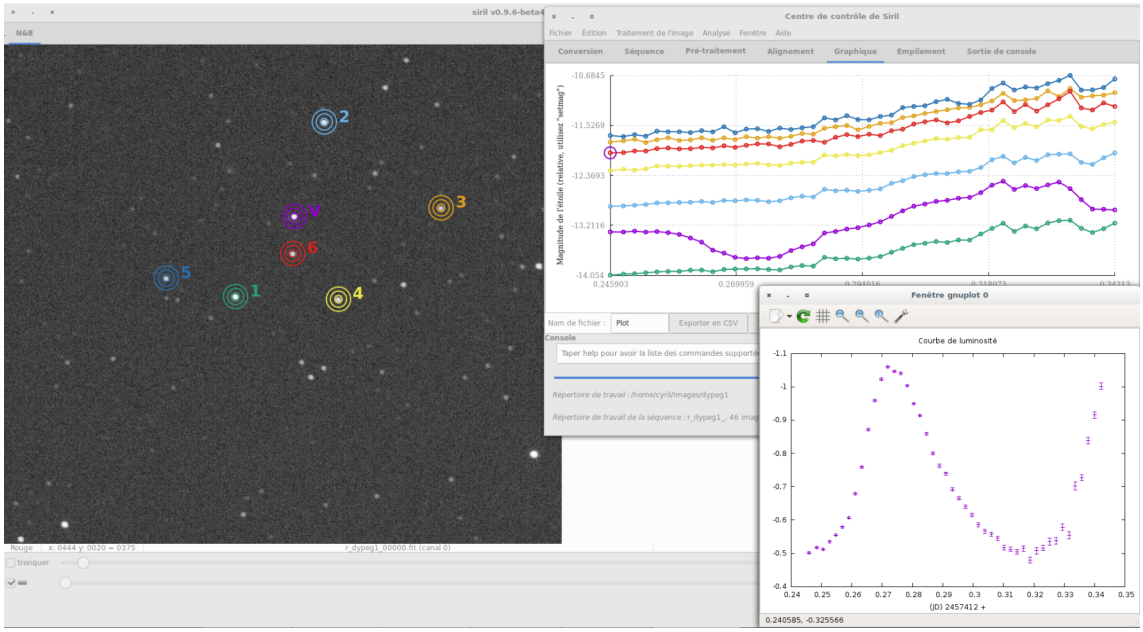
X axis: Julian Day

- format: geocentric based on UTC

Y axis: magnitude difference:

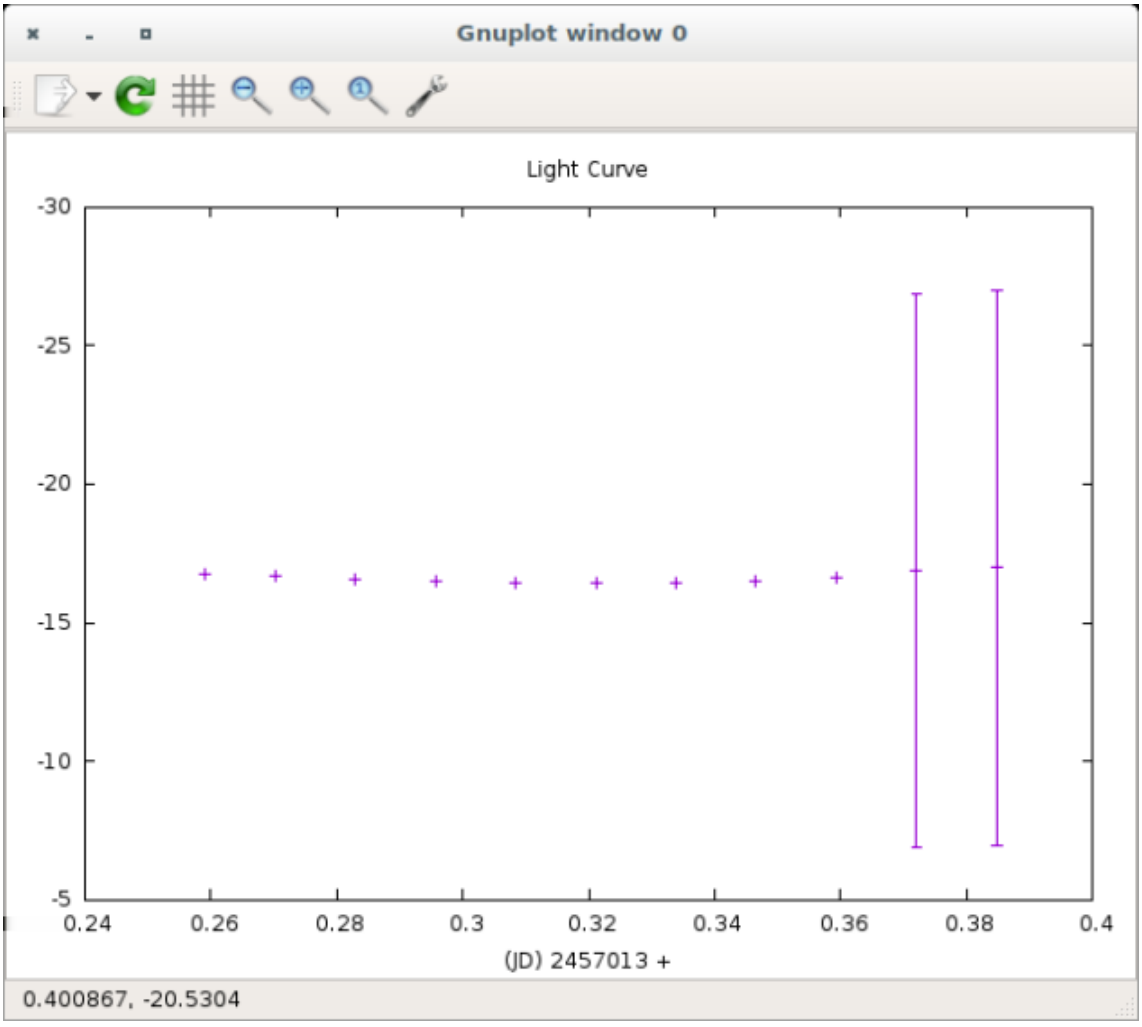
- Magnitude of the variable - magnitude of the super star.
- The super star is the star constructed from the stars of reference.

Result



General comments

An error may occur



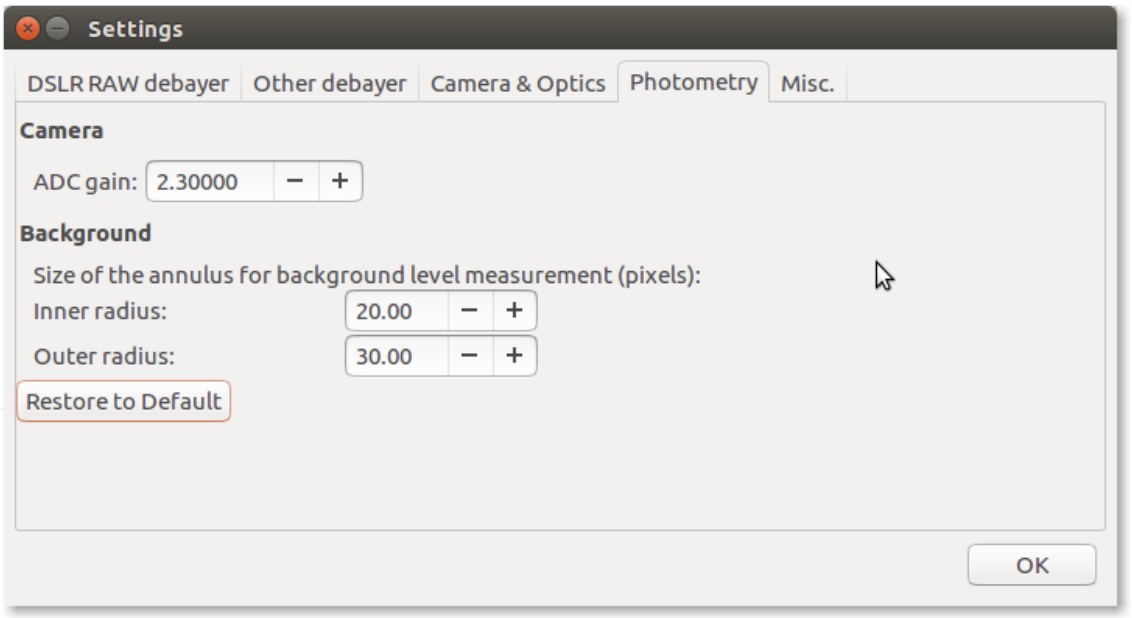
For an unknown reason, the error on the magnitude could not be determined and is therefore given the value 9.99

> We calculate the magnitude with the old algorithm much less reliable but that can not fail.
> The error remains at 9.99.

You have to understand why: look at the console

```
12:50:14: Inner and outer radii are too small. Please update values in setting box.
12:50:14: Inner and outer radii are too small. Please update values in setting box.
```

Change settings and try again

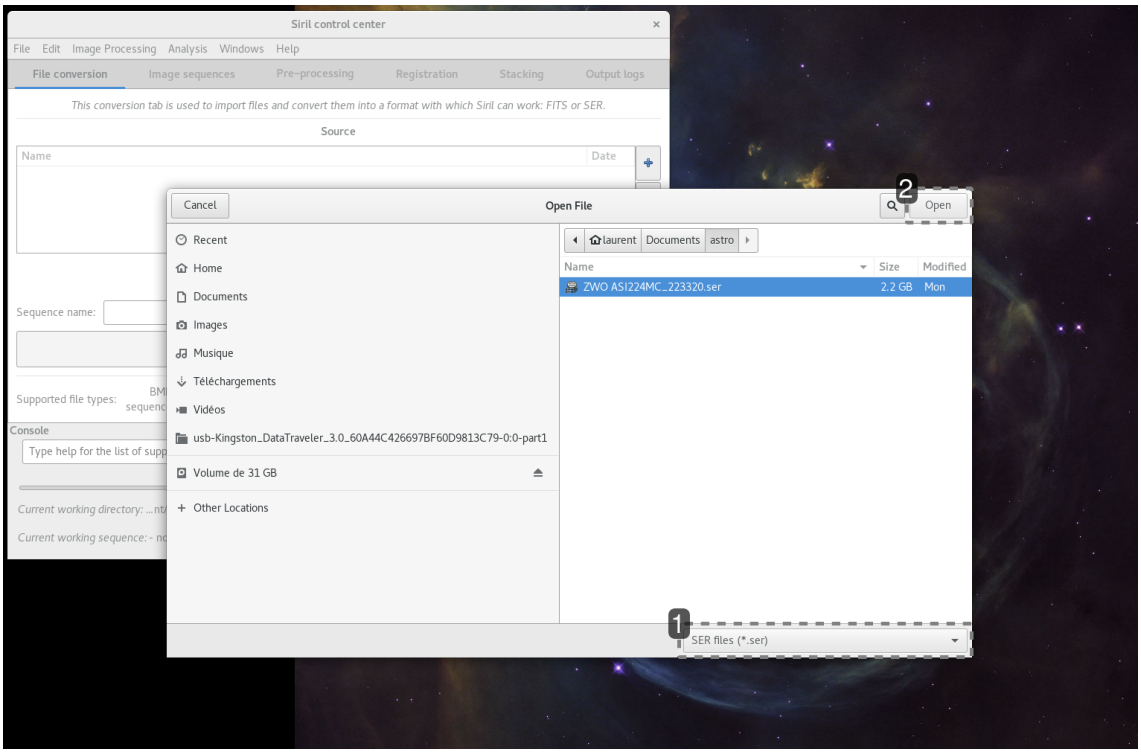


2.2. Planetary

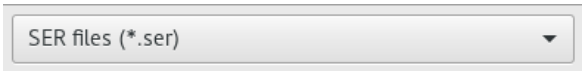
2.2.1. Use SER files

Open the file

Menu ☰File/Open File

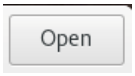


1

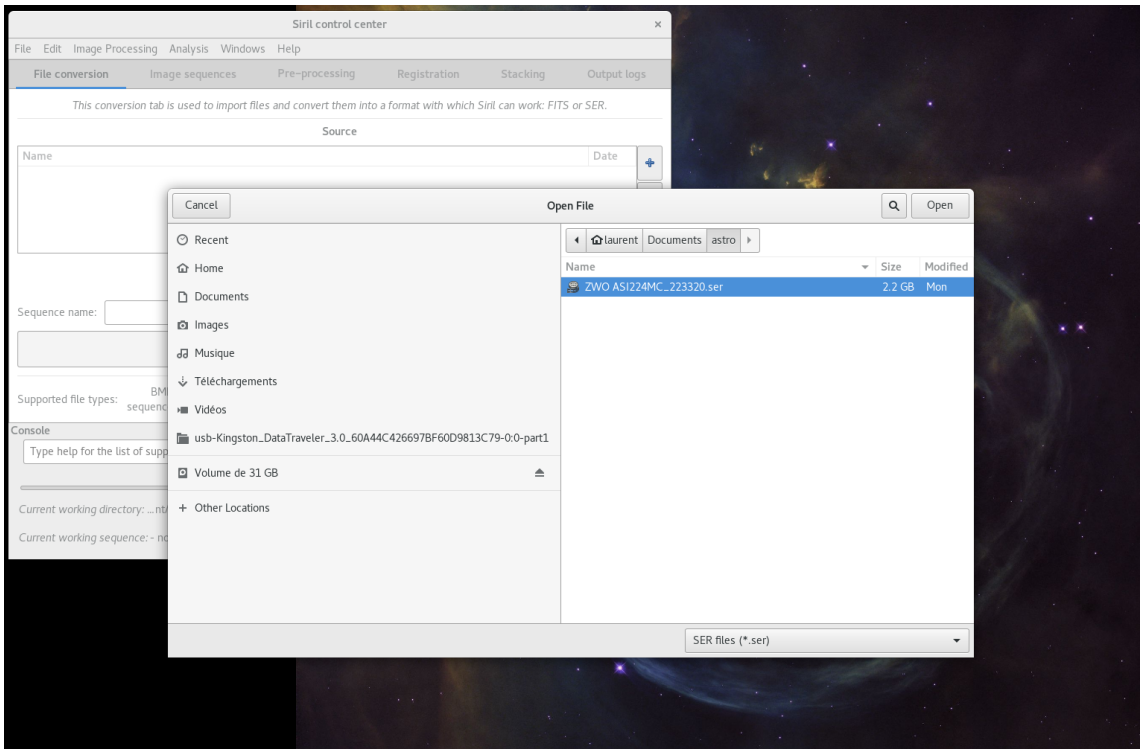


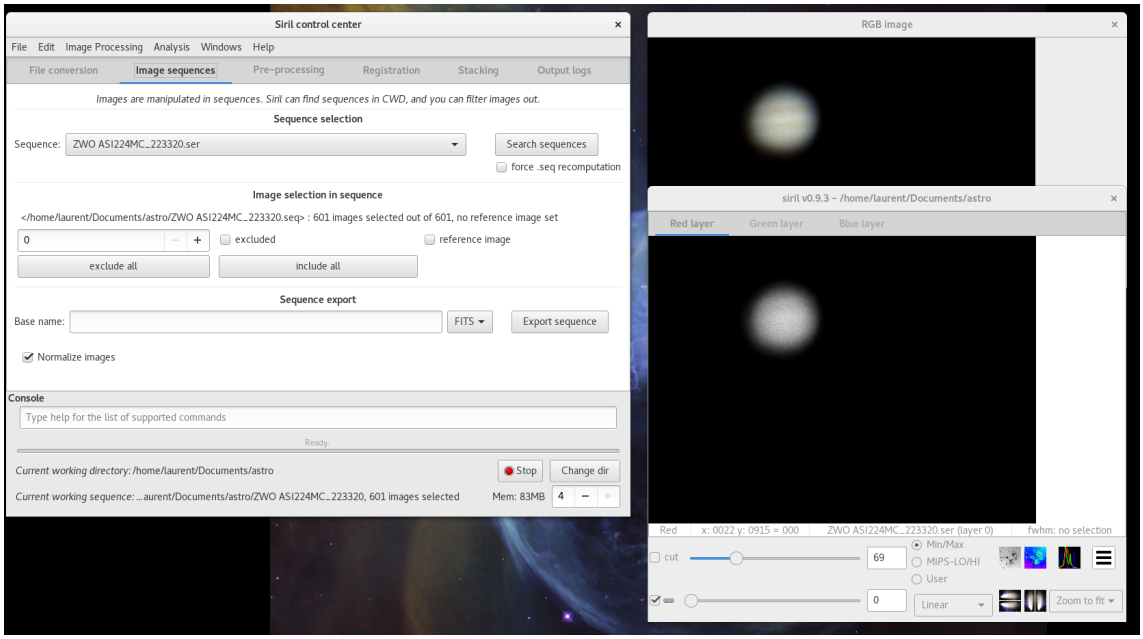
Select SER files (* .ser) from the dropdown menu

2



Click on Open once the file selected





The SER files are converted on the fly in sequences

2.2.2. Usual planetary processing (avi)

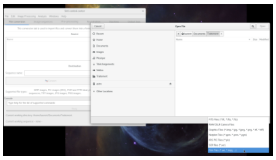
Avi planetary processing step by step

> Support for films such as AVI has been dropped and many functions of Siril will not work for them, in favour of the SER format. This version of Siril provides a way to convert films into SER sequences, on which all Siril processing functions are supported.

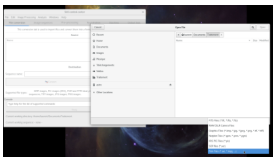
1. open file

   Go to the File/Open menu

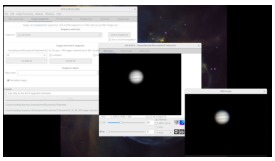
   In the file type selection list choose Film Files (*.avi,*.mpg,...)



2. File select

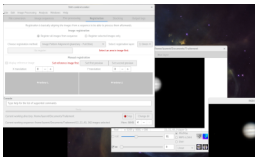


   Select the AVI file to be processed and click Open



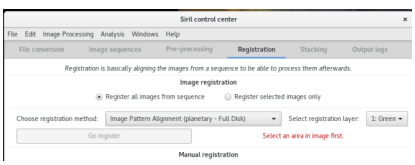
3. reference image

   Go to the Registration tab



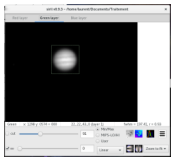
4. Registration

   Set Image Pattern Alignement (Planetary - Full Disk) as Registration method



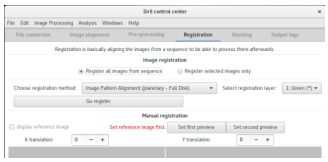
5. Sélection

Go to the preview window and select the area around the planet

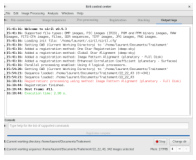


6 Image de référence

Click on Go registrer



In the Output tab logs will point you to the best picture

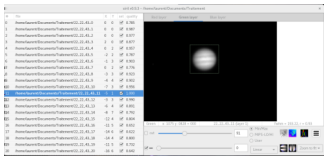


Best frame : here **#11**

7 reference image

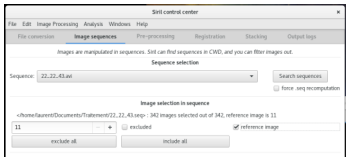
Go to the preview window, display the image list ☰

Select the best picture with a double click



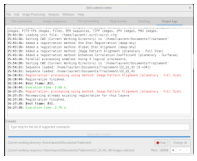
8 reference image selection

Go to the tab Images sequences then check reference image



9 Registration

Go to the tab Registration and click on Go register



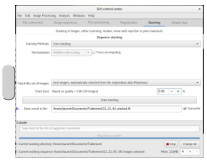
10 Stacking

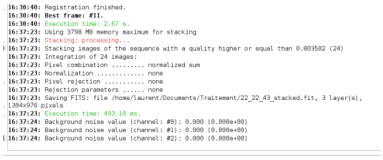
Use Sum stacking with avi FILES

Choose best images, automatically selected from registration data (Planetary)

Choose the % of images to use, depending on the quality level referred

Click on Start stacking



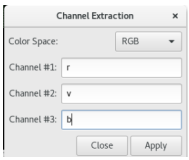


11 Extraction of RGB channels

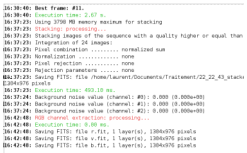
Go to the Image menu ☰Processing/Extraction/Channel...

Choose r in Channel #1

v in Channel #2 et b in Channel #3



Click on apply



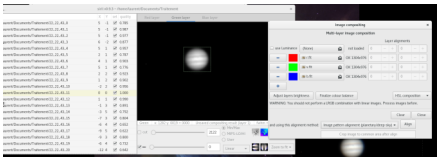
12 Image compositing

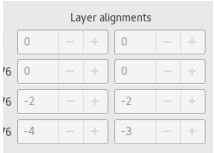
Go to the ☰Image Processing/RGB Compositing

Choose files previously created for red, green and blue

Choose Image pattern alignment (planetary/deep-sky)

Select an area around the planet in the pre viewing window and then click on Align

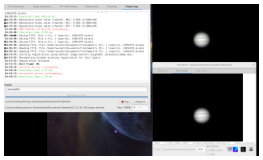




Cliquez sur Close

13 flip your image if needed

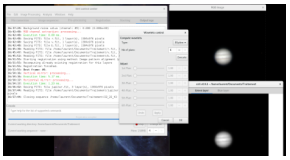
Then save it



14 wavelet processing

Load your saved image

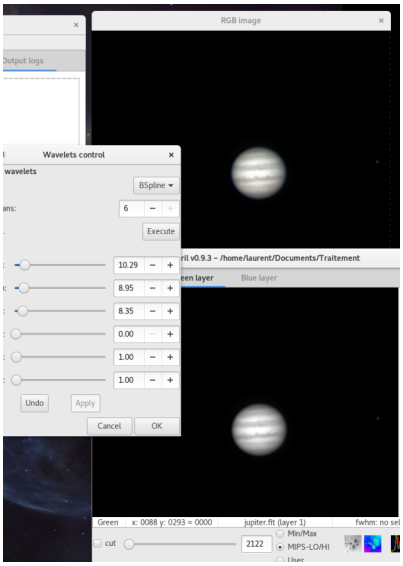
Menu ☰Image Processing/ A trous Wavelets Tranform...



15 Traitement

Click on Execute

Make your settings



16 Save file

Click on ok close the wavelet dialog box

Menu ☰file/Save as... and choose the saving format

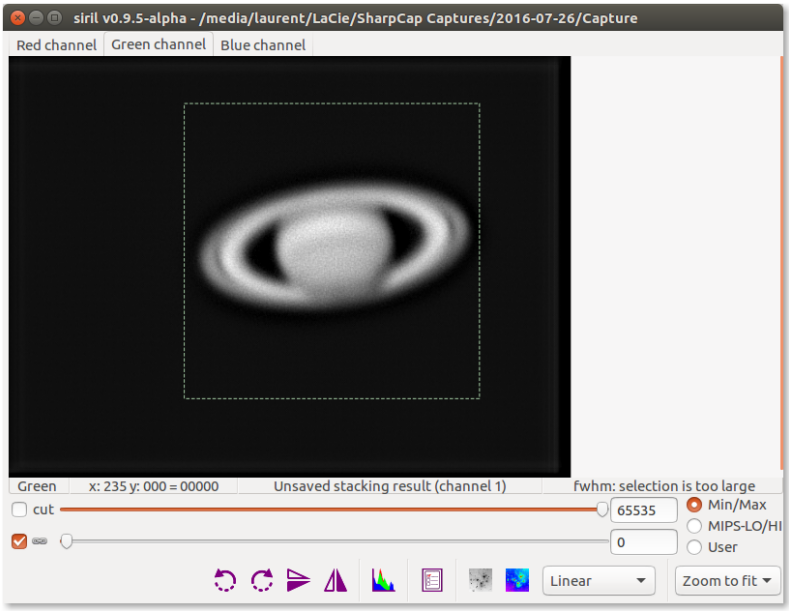
2.2.3. Align RGB channels

At the end of the stack, Siril provide an option to automatically align RGB channels.

Automatically align RGB channels

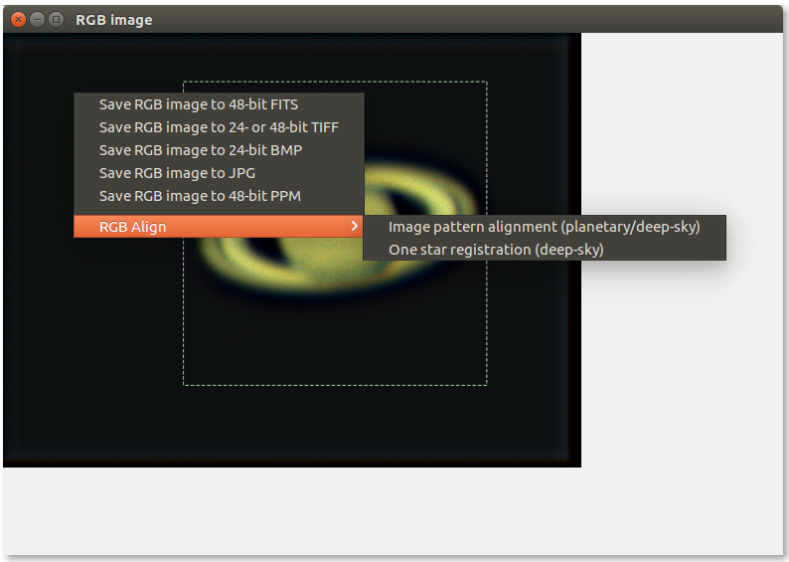
1. At the end of the stack,

   select an area in gray scale image windows



2. Then in color image windows

   Right click to open pop up menu


RGB image
Save RGB image to 48-bit FITS
Save RGB image to 24- or 48-bit TIFF
Save RGB image to 24-bit BMP
Save RGB image to JPG
Save RGB image to 48-bit PPM
RGB Align
Image pattern alignment (planetary/deep-sky)
One star registration (deep-sky)

3. Select RGB align
4. Choose Image pattern alignment or one star registration.

   In most of the cases use Image pattern alignment.

   You can align with a star or a Jupiter's satellite for example with one star registration

Result

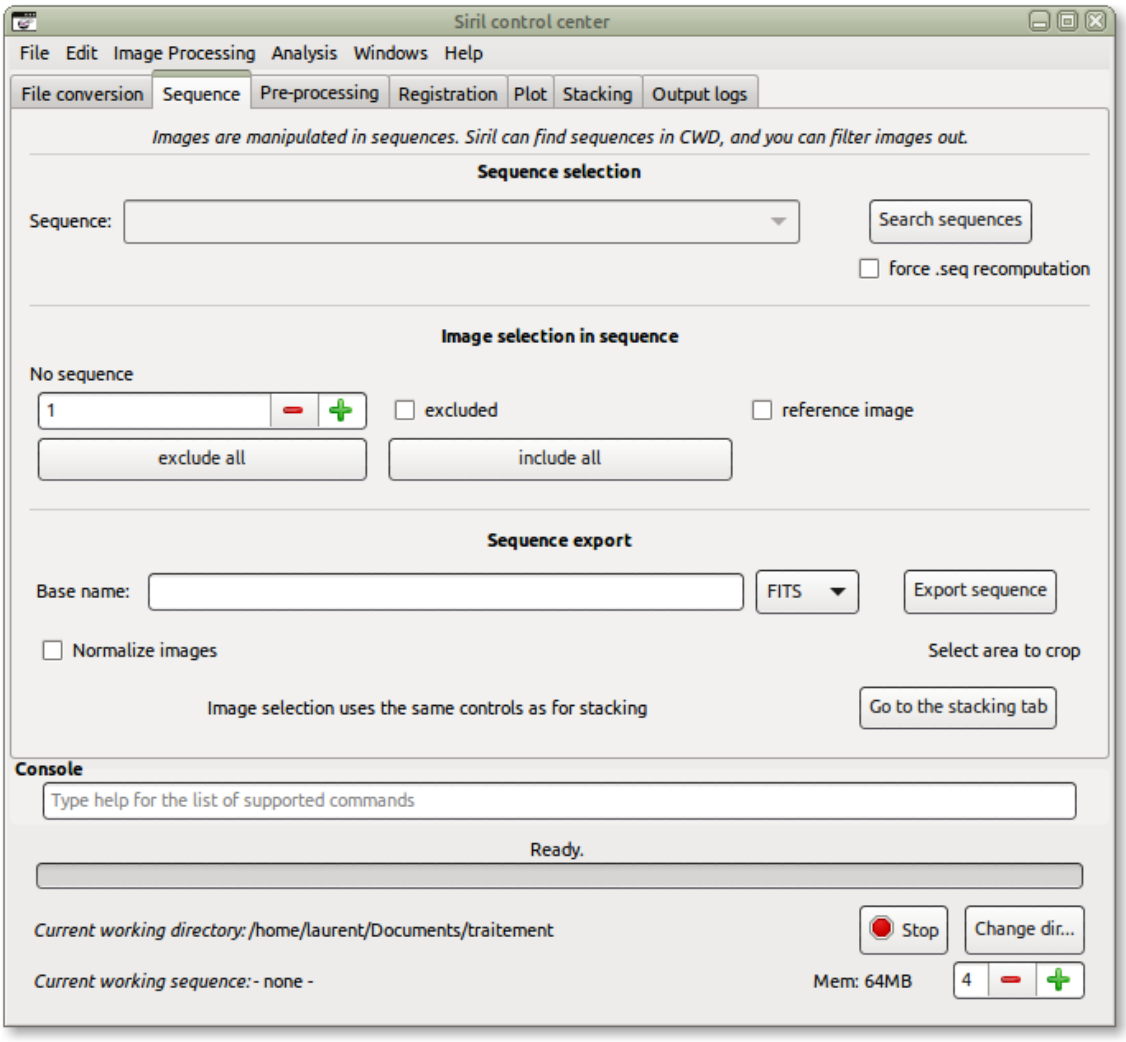
The RGB channels alignment is done

```
11:32:25: Recomputing already existing registration for this layer
11:32:26: Registration finished.
11:32:26: Best frame: #0.
```

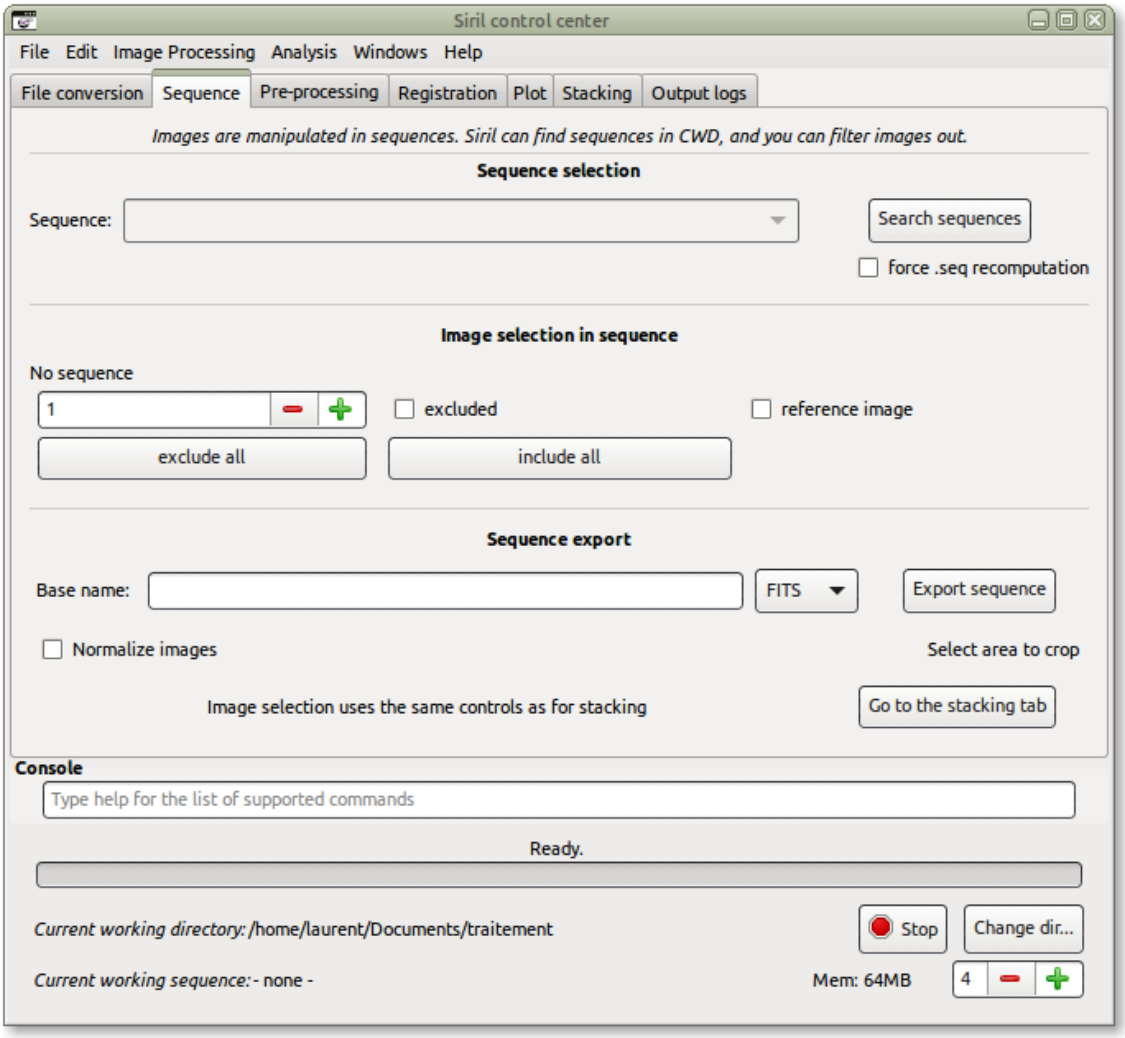
2.2.4. Archiving planetary videos

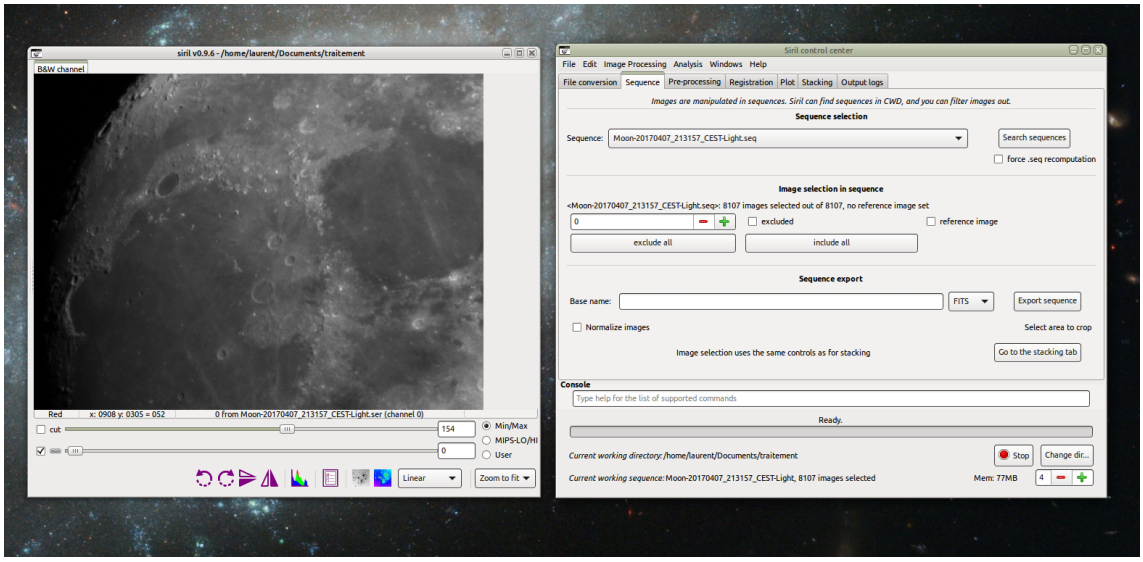
Siril allows you to export a user-defined percentage of the best images from a SER video.

1. Go to the Sequences tab

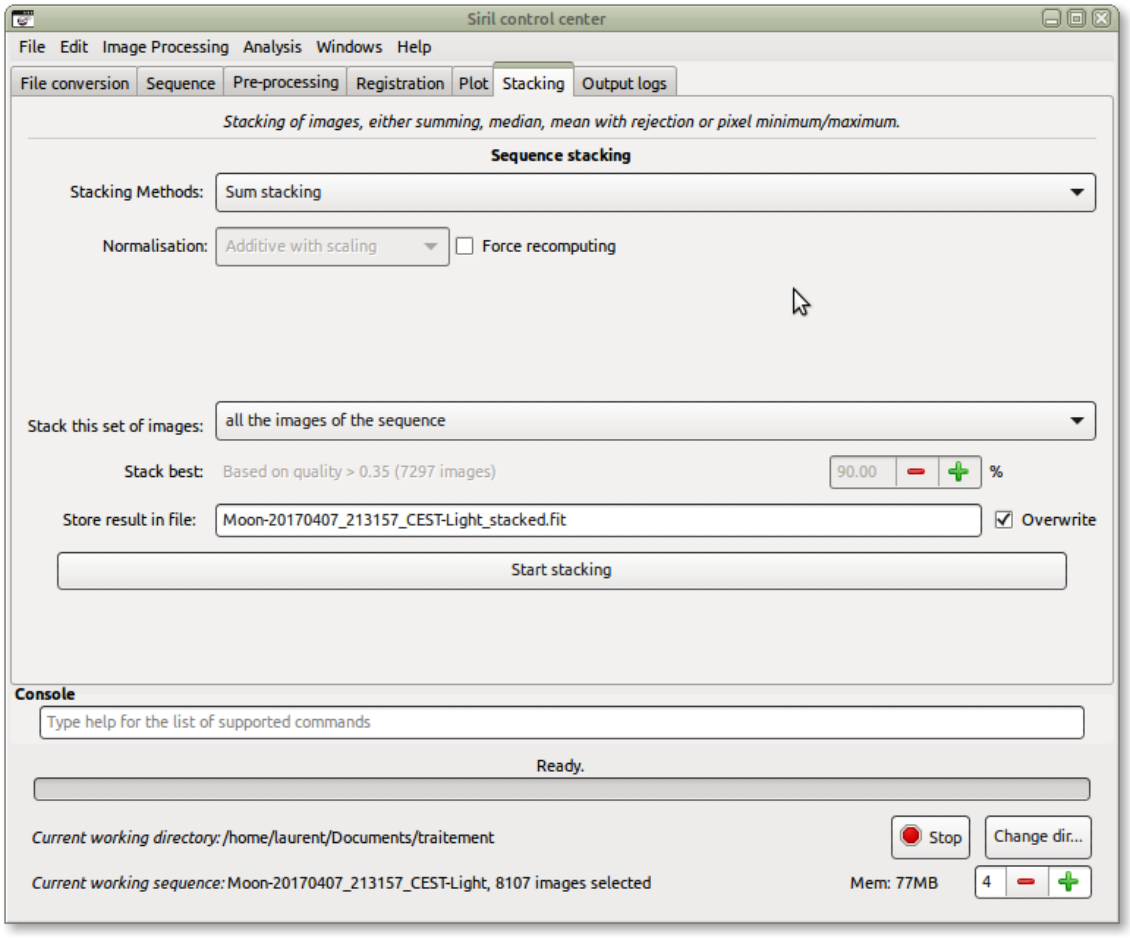


2. Select the file **Ser** or Avi to archive

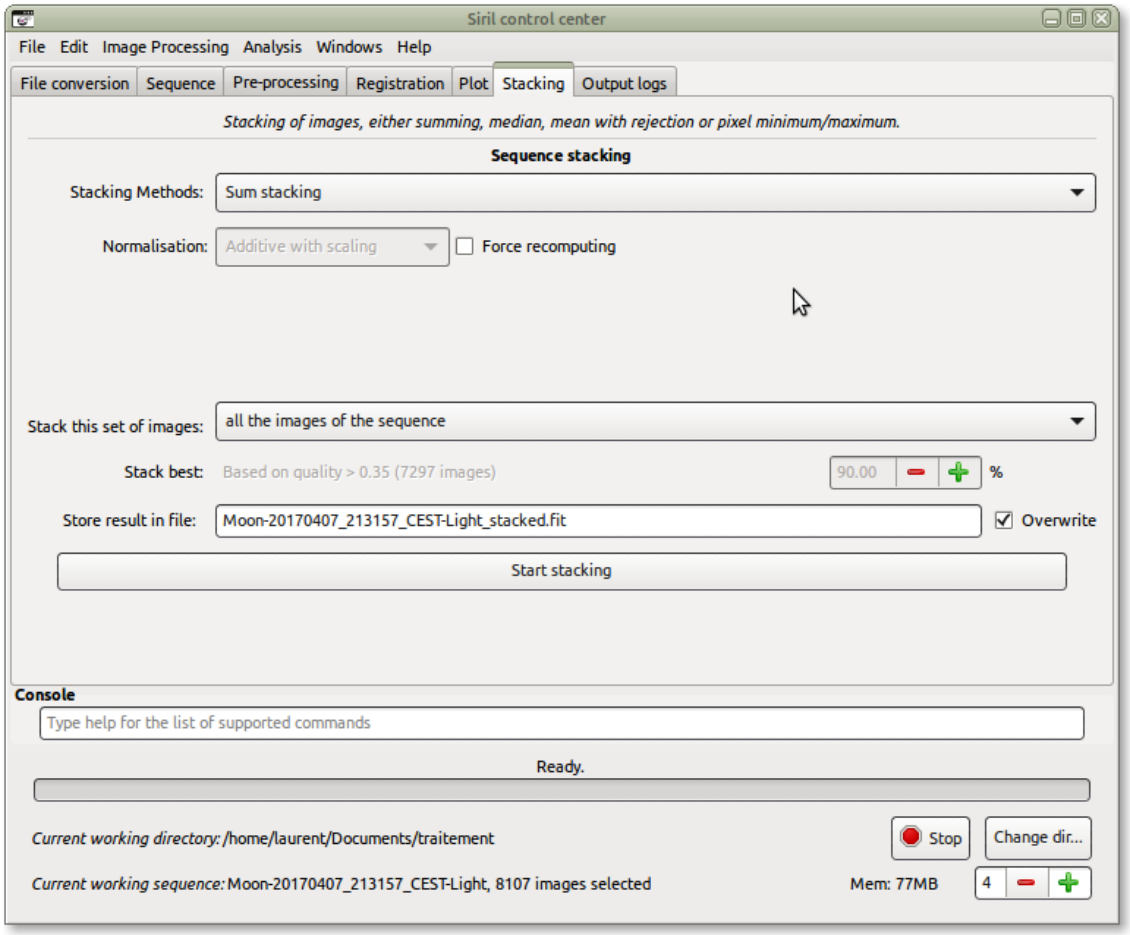




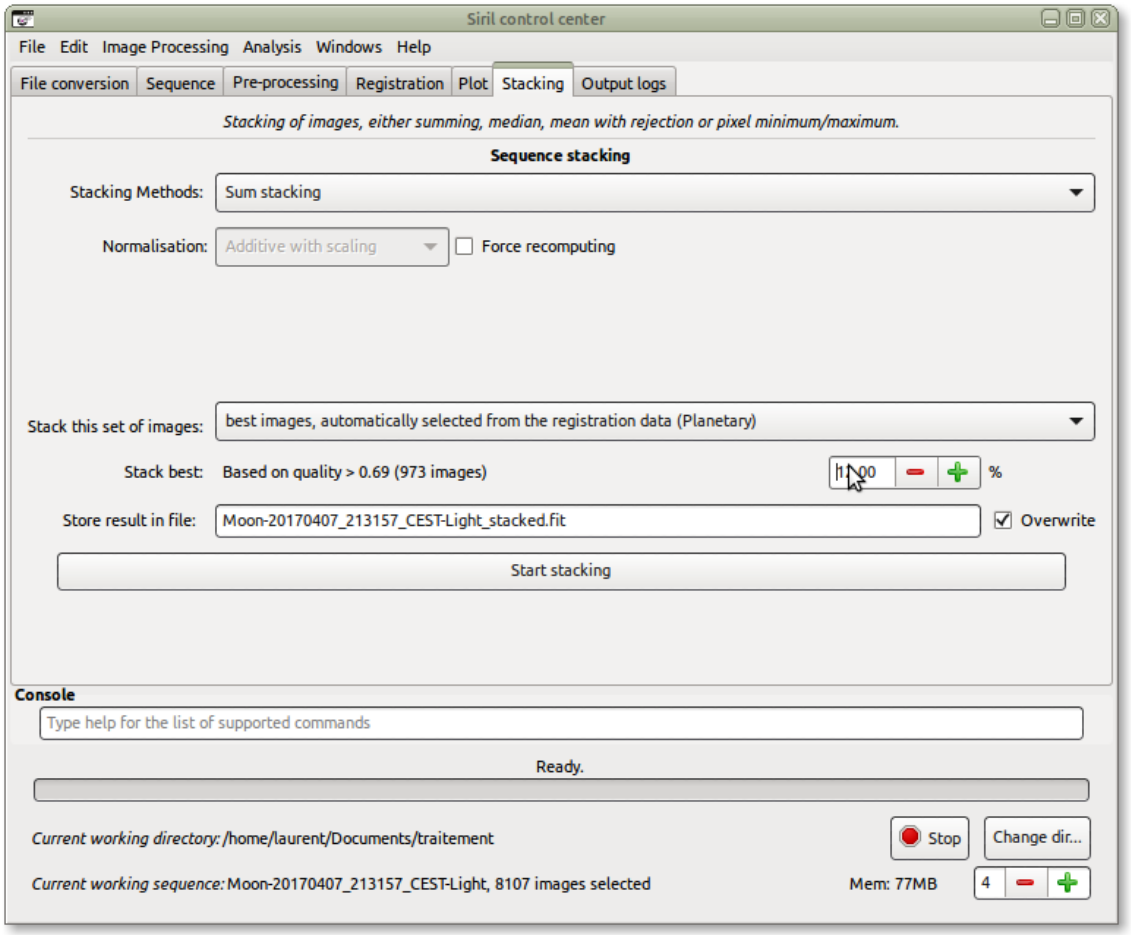
3. Register your sequence if you have not already done.
4. Go to the Stackingt tab


Siril control center
File Edit Image Processing Analysis Windows Help
File conversion Sequence Pre-processing Registration Plot Stacking Output logs
Stacking of images, either summing, median, mean with rejection or pixel minimum/maximum.
Sequence stacking
Stacking Methods: Sum stacking
Normalisation: Additive with scaling
Force recomputing
Stack this set of images: all the images of the sequence
Stack best: Based on quality > 0.35 (7297 images)
90.00
%
Store result in file: Moon-20170407_213157_CEST-Light_stacked.fit
Overwrite
Start stacking
Console
Type help for the list of supported commands
Ready.
Current working directory: /home/laurent/Documents/traitement
Stop
Change dir...
Current working sequence: Moon-20170407_213157_CEST-Light, 8107 images selected
Mem: 77MB
4

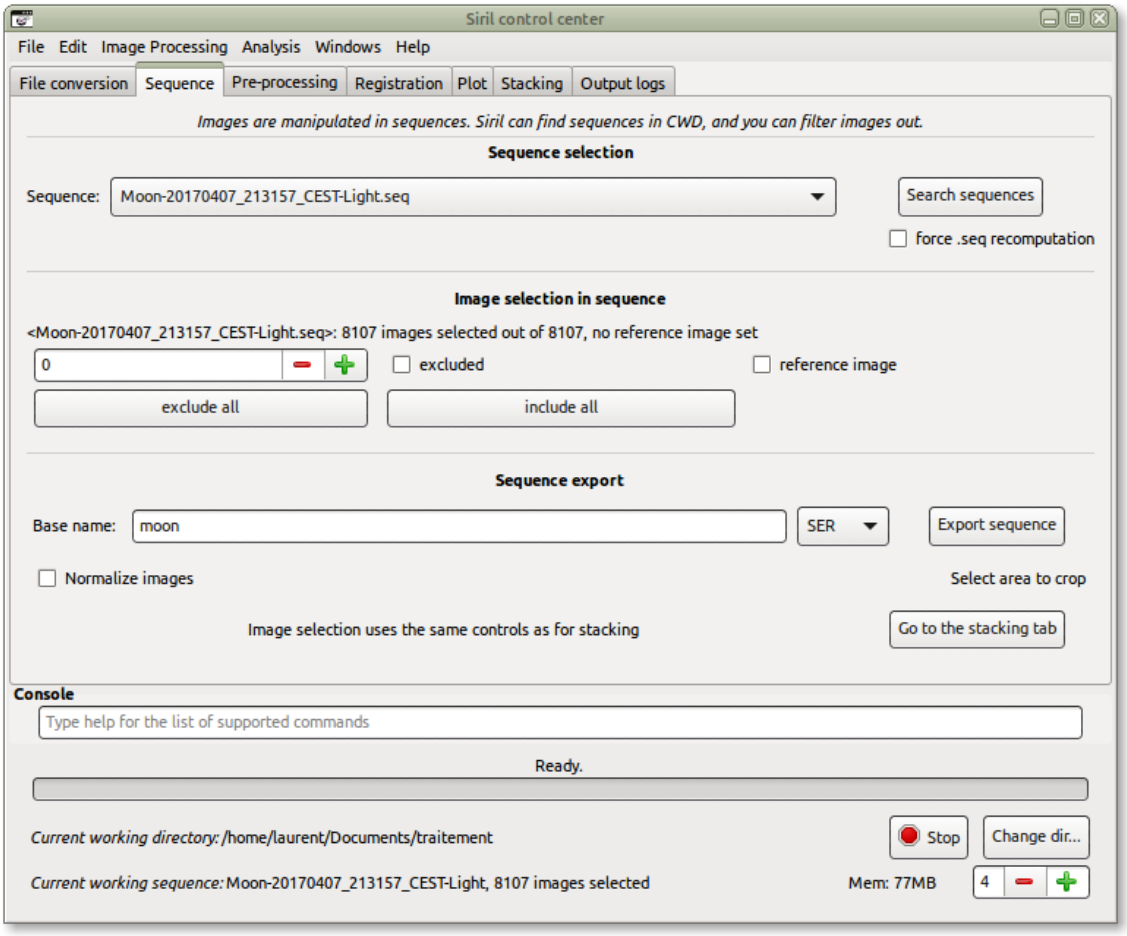
5 *Choose best images, automatically selected from the alignment information (Planetary)*


Siril control center
File Edit Image Processing Analysis Windows Help
File conversion Sequence Pre-processing Registration Plot Stacking Output logs
Stacking of images, either summing, median, mean with rejection or pixel minimum/maximum.
Sequence stacking
Stacking Methods: Sum stacking
Normalisation: Additive with scaling
Force recomputing
Stack this set of images: all the images of the sequence
Stack best: Based on quality > 0.35 (7297 images) 90.00 %
Store result in file: Moon-20170407_213157_CEST-Light_stacked.fit
Overwrite
Start stacking
Console
Type help for the list of supported commands
Ready.
Current working directory: /home/laurent/Documents/traitement
Stop
Change dir...
Current working sequence: Moon-20170407_213157_CEST-Light, 8107 images selected
Mem: 77MB
4

6 Choose the quality of images to keep

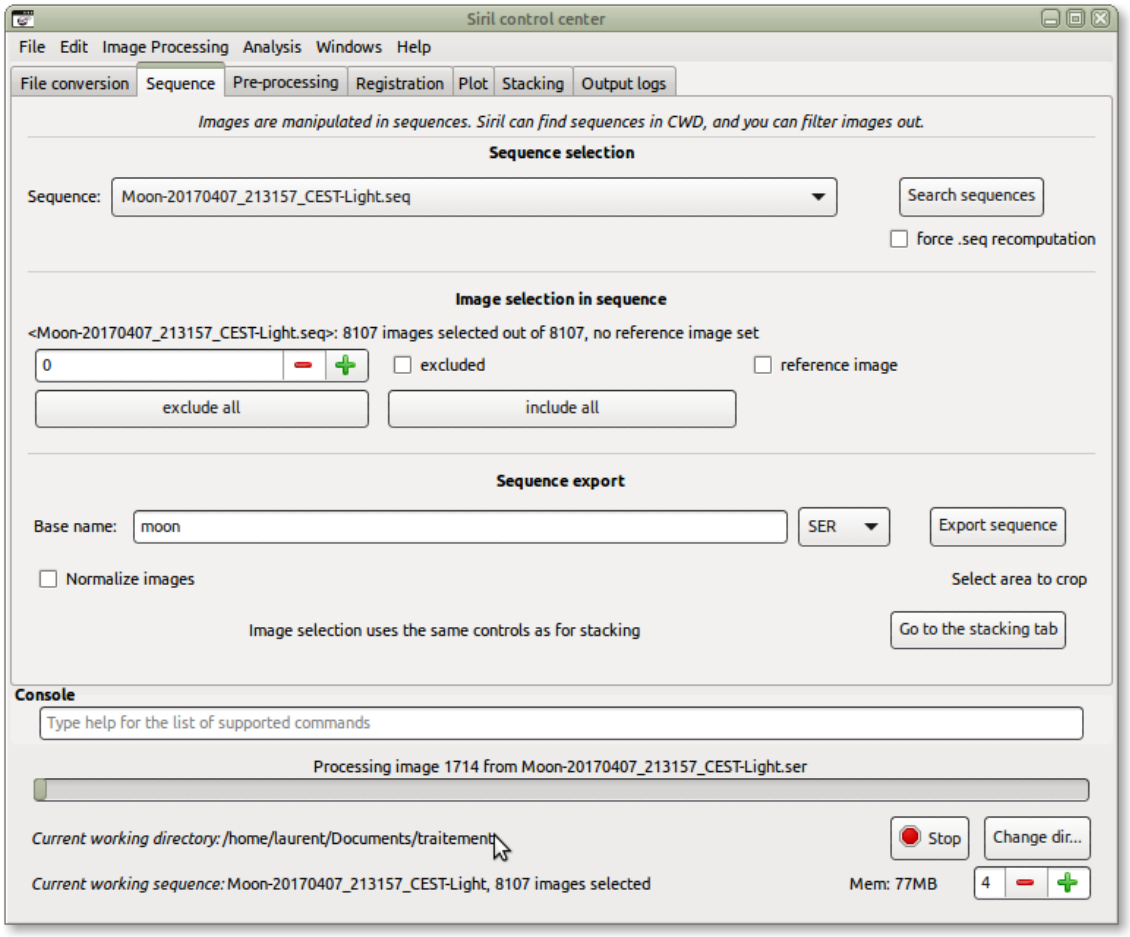

Siril control center
File Edit Image Processing Analysis Windows Help
File conversion Sequence Pre-processing Registration Plot Stacking Output logs
Stacking of images, either summing, median, mean with rejection or pixel minimum/maximum.
Sequence stacking
Stacking Methods: Sum stacking
Normalisation: Additive with scaling
Force recomputing
Stack this set of images: best images, automatically selected from the registration data (Planetary)
Stack best: Based on quality > 0.69 (973 images)
100
%
Store result in file: Moon-20170407_213157_CEST-Light_stacked.fit
Overwrite
Start stacking
Console
Type help for the list of supported commands
Ready.
Current working directory: /home/laurent/Documents/traitement
Stop
Change dir...
Current working sequence: Moon-20170407_213157_CEST-Light, 8107 images selected
Mem: 77MB
4

7. Back to the `Sequences` tab
8. Enter a name in `Base Name`
9. Choose the **SER** format to export the sequence
10. Click Export Sequence


Siril control center
File Edit Image Processing Analysis Windows Help
File conversion
Sequence
Pre-processing
Registration
Plot
Stacking
Output logs
Images are manipulated in sequences. Siril can find sequences in CWD, and you can filter images out.
Sequence selection
Sequence:
Moon-20170407_213157_CEST-Light.seq
Search sequences
force .seq recomputation
Image selection in sequence
<Moon-20170407_213157_CEST-Light.seq>: 8107 images selected out of 8107, no reference image set
0
excluded
reference image
exclude all
include all
Sequence export
Base name:
moon
SER
Export sequence
Normalize images
Select area to crop
Image selection uses the same controls as for stacking
Go to the stacking tab
Console
Type help for the list of supported commands
Ready.
Current working directory: /home/laurent/Documents/traitement
Stop
Change dir...
Current working sequence: Moon-20170407_213157_CEST-Light, 8107 images selected
Mem: 77MB
4

Result

The sequence is exported



You will find the file in the current working directory

3. Authors

Vincent[https://free-astro.org/index.php/User:Vincent] is a computer scientist (PhD) and uses Siril as an amateur to process images from a Canon EOS, a QSI and a Basler camera on a 410mm telescope.

Cyril[https://free-astro.org/index.php/User:Lock42] is a physicist (PhD), now IT engineer at CNRS motivating new developments and providing high quality processing algorithms to Siril.
Laurent is a senior IT project manager and also uses Siril to process images from a Canon EOS and a ZWO camera on a 250mm telescope. He manages the formatting of the documentation, using Scenari[https://scenari.org/] technology.

François Meyer wrote the initial (up to v0.8) versions. Here are the legacy Sourceforge project[http://sourceforge.net/projects/siril/] and website[http://dulle.free.fr/alidade/siril.php?lan=en].

See the AUTHORS[https://free-astro.org/svn/siril/trunk/AUTHORS] file for the complete list of contributors.

Glossary

2D Gaussian	$G(x,y)=B+Ae^{-(\frac{(x-x_0)^2}{2\sigma_x^2}+\frac{(y-y_0)^2}{2\sigma_y^2})}$
2D Gaussian	$G(x,y)=B+Ae^{-(\frac{(x-x_0)^2}{2\sigma_x^2}+\frac{(y-y_0)^2}{2\sigma_y^2})}$
A BIAS frame *= offset*	is an image taken with the shutter closed and the shortest possible exposure time. Basically it corresponds to an exposure of 1/4000s with modern DSLRs. The BIAS shows the electronic noise and readout signal of the camera and the more BIAS exposures are used for the calculation of the master BIAS, the less noise will be introduced into the corrected images. Note that BIAS frames MUST be taken at the same ISO than light frames.
Adaptive Homogeneity-Directed *= AHD*	The direction to interpolate is chosen to minimize the level of color artifacts. Due to the rectangular sampling lattice in Bayer pattern, interpolation is performed in horizontal and vertical directions only . Directional interpolation uses filterbank techniques to cancel aliasing from CFA sampling.
addmax	`addmax filename` addmax compute a new image IMG with IMG_1 and IMG_2. The pixel of IMG_1 is replaced by the pixel at the same coordinates of IMG_2 if the intensity of 2 is greater than 1.
ADU *= Analog Digital Unit* *= Binary Number* *= Digital Count*	There are several concepts and issues going on here that are getting mixed together. **First the ADC** An ADC (Analog To Digital Convertor) takes signal values in a certain range and converts then to a digital number. Let's run some numbers:

A N bit ADC will be able to represent 2^N states. So a 4 bit ADC can represent 16 distinct states, 8 bits => 256 States and a 12 bit convertor will give you 4096 states. By itself that tells you nothing, you need to know that going from a digital representation of say 533 -> 534 means that the input signal changed by certain amount. To keep the example simple I'm going to say that I have an input range from 0 Volts to 4.095 Volts on a 12 bit convertor, that means that when I have a digital number of 0 then I must have had 0 on the input. Let's also assume everything is ideal for now. With my sneaky choice of input range you can see that for every change in the ADC output digital number by one count then the input must have changed by 1 mV.

Why did I pick 4.095? Well a 12 bit convertor will have 4096 digital values that range from 0 to 4095 (2^N-1 when all the bits are set to "1"). So 4.095 Volts/4095 = 1 mv/count. That means that if you have a 3.567 Volt value on the output and it changes to 3.568 Volts (1 mV change on the input) then the digital number on the output should change from 3567 to 3568 digital number on the output from 110111101111 to = 110111110000 in raw binary.

Each of these binary steps are called an ADU (Analog Digital Unit) or Binary Number or Digital Count etc. etc. to keep it separate from the number of bits that the ADC has. This is to ensure that the relationship is in a linear range. If you start using "bits" then it is in a logarithmic range (2^N). The main reason these other names/units are used is that LSB's used to be used. What is a LSB? - it stands for Least Significant Bit, i.e. the zeroth bit in the digital number, which changes precisely in step with the ADU but using LSB (and the implied 'bit") is confusing. Imagine saying, " the ADC changed by 30 LSB's therefore the light flux must have changed by 3000 photons ". 30 LSB's really doesn't make sense, it is a mixing of terminology, so people are moving away from that. But that has meant that many different terms being used instead confusing again...

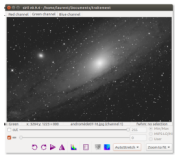
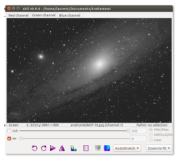
So you have a conversion from voltage into a digital number.

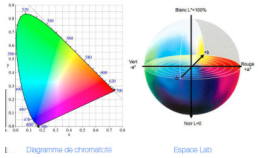
But you are collecting photons, where does voltage come from?

Next the Voltage conversion

All image sensors at some point in the signal chain will convert the collected signal charge into voltage to interact with the outside world. This is always done using a capacitance and a buffer amplifier. This capacitor and the gain of the amplifier set the conversion from electrons into voltage as A/C_sense which is derived from the Q= CV equation. With C_sense = the sense node capacitance and A = gain of the buffer amplifier.

Next the photon conversion

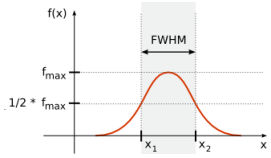
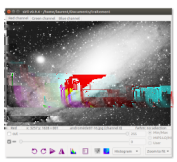
	When a Photon comes rattling into a pixel it has to make it into the photosensitive substrate, this interface is controlled by the fresnel equations and dictates how much light goes into the substrate based upon the indices of refraction and the various materials and AR coatings present (AR = Anti-reflection). This is known as the external QE (Quantum efficiency). Once in the substrate the photons are absorbed through it is depth and generate electron/hole pairs through the photo electron effect. Electric fields sweep the carriers into storage structures. The absorption depth and the depth of influence of the electric fields (these two depths may not be co-incident) are what determines the Internal QE. NOt every converted photon's carriers get collected. These collected carriers (either electrons or holes) are brought to the sense node and converted into voltages. Where this conversion happens (carrier to voltage) is determined by the type of sensor, a CCD or CMOS image sensor. The total conversion process; (Photon flux) * (Integration period) * QE_external *QE_Internal * A /C_sense * ADC_conversion Will give you units of Photons/ADU. (1*[http://physics.stackexchange.com/a/149550]*)
Aperture photometry	Aperture photometry is a technique concerned with measuring the flux of an astronomical object.
Asinh	The inverse hyperbolic sine is commonly used, it reproduces the perceptive capacity of the human eye, allowing to perceive significantly different brightness levels simultaneously. The asinh function is close to the logarithmic mode but has a better behavior around zero. 
AutoStretch	Siril performs auto stretching curves to adjust the image and make it visible on the screen. 
Banding	Phenomenon manifested as horizontal bands of darker than the rest of the image. This defect is easily visible on a plain and clear background.
bg	`bg` Returns the background level of the image loaded in memory.

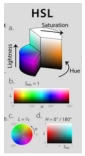
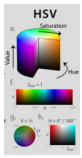
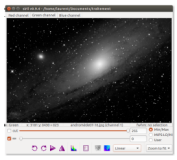
bgnoise	`bgnoise` Returns the background noise level.
BT.709 *= Rec. 709*	Standardizes the format of high-definition television, having 16:9 (widescreen) aspect ratio. The first edition of the standard was approved in 1990.
cd	`cd directory` Defined the new working *`directory`*
cdg	`cdg` Return the coordinates of the center of gravity of the image.
CFA *= Color Filter Array*	The CFA image is a black and white image that shows the signal received by each pixel of the sensor. The pixels are alternately covered with red filters, green and blue. In general, there are twice as many green pixels than red and blue (the case of the Bayer matrix). For example, if the observed object is uniformly red, only the red pixels will be illuminated.
CIE L*a*b* *= CIELAB*	**CIE L*a*b*** *[https://fr.wikipedia.org/wiki/CIE_L*a*b*]*, often abbreviated CIELAB is a color space for surface colors. • **The component L*** is clarity, which ranges from 0 (black) to 100 (white). • **The component a*** represents a range of 600 levels on a red pin (299 positive) → green (-300 negative value) via the gray (0). • **The component b*** represents a range of 600 levels on a yellow axis (299 positive) → blue (negative value -300) through the gray (0). 
clear	`clear` Clears the graphical output logs.
clearstar	`clearstar` Clear all the stars saved in memory and displayed on the screen.
close	`close` Closes the open image and the open sequence correctly, if necessary.

convertraw	`convertraw basename [-debayer]` Convert DSLR RAW into Siril's FITS images. The argument `basename` is the basename of the new sequence. The option `-debayer` applies demosaicing to images
cosme	`cosme filename` Apply the local mean to a set of pixels on the in-memory image (cosmetic correction). The coordinate of this pixels are in an ASCII file [list file]. COSME is adapted to correct residual hot and cold pixels after preprocessing. The line P x y type will fix the pixel at coordinates (x, y) type is an optional character (C or H) specifying to Siril if the current pixel is cold or hot. This line is created by the command find_hot but
cosme_cfa	`cosme_cfa filename` Same function that COSME but applying to *`RAW CFA images.`*
crop	`crop x, y, width, height` Crops the current image within the rectangle previously selected.
Ctrl+B	Saving the BMP image
Ctrl+J	Saving the JPG image
Ctrl+O	Open file
Ctrl+P	Saving the NetPBM image
Ctrl+S	Saving the image (FIT format)
Ctrl+T	Saving the image (TIFF format)
Ctrl+Z	Undo
cut	☑Cut Instead of keeping pixels with values greater than the value "hi" white when checked: displays black pixels when saturated
Darks	Dark frames are made at the same exposure time and ISO as the subject light frames but in the dark: use your lens/telescope cap or close the shutter for example. They contain the thermal noise associated with the sensor, the noise being proportional to temperature and exposure time. Hence, they should be made at approximately the same temperature as the light frames, this is the reason why we make dark frames at the end, or in the middle of the imaging session. Like with the BIAS frames, the more dark exposures are used for the calculation of the master dark, the less noise will be introduced into the corrected images.

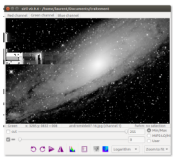
ddp	`ddp level coef sigma` Performs a DDP (digital development processing) as described first by Kunihiko Okano*[https://www.sbig.com/astronomy/hall-of-fame/kunihiko-okano/]*. This implementation is the one described in IRIS. It combines a linear distribution on low levels (below level) and a non-linear on high levels. It uses a Gaussian filter of sigma sigma multiplies the resulting image by coef. The typical values for sigma are included between 0.7 and 2
Dynamic PSF	Dynamic PSF is a dynamic tool inspired by the PixInsight routine of the same name. It is used to fit unsaturated stars within the image.
entropy	`entropy` Computes the entropy of the opened image on the displayed layer, only in the selected area if one has been selected or in the whole image else.
exit	`exit` Quits the application.
extract	`extract NbPlane` Extracts NbPlane Planes of Wavelet domain.
fdiv	`fdiv filename scalar` Divides the image in memory by the image given in argument. The resulting image is multiplied by the value of the scalar argument. Please check that the image is in the working directory
fftd	`fftd modulus phase` Applies a Fast Fourier Transform to the image loaded in memory. Modulus and phase given in argument are saved in FITS files.
ffti	`ffti modulus phase` This function is used to retrieve corrected image applying an inverse transformation. The modulus and phase used are the files given in argument.
fill	`fill value x y width height` Fills the whole current image (or selection) with pixels having the value intensity.
fill2	`fill2 value x y width height` Same command than fill but this is a symmetric fill of a region defined by the mouse. Used to process an image in the Fourier (FFT) domain.
find_cosme	`find_cosme cold_sigma hot_sigma` Applies an automatic detection of cold and hot pixels following the thresholds written arguments

find_cosme_cfa	`find_cosme_cfa cold_sigma hot_sigma` Same command than `find_cosme` but for monochromatic CFA images.
find_hot	`find_hot filename cold_sigma hot_sigma` The command provides a file lists "filename" (format text) in the working directory which contains the coordinates of the pixels which have an intensity "hot_sigma" times higher and "cold_sigma" lower than standard deviation. We generally use this command on a master-dark file.
findstar	`findstar` Detects stars having a level greater than a threeshold computed by Siril.
fixbanding	`fixbanding amount sigma` Try to remove the canon banding. "Amount" define the amouont of correction. Sigma defines a protection level of the algorithm, higher sigma gives higher protection.
Flat	Telescopes usually do not illuminate the detector homogeneously. Moreover, dust on optical surfaces and sensor lead to darker patterns on resulting image, and the sensor itself has a different response to the number of photons hitting different photosites. To correct these effects one need to divide each light images by the master flat, which should be the median of single exposures of a homogeneously and non-saturating illuminated area.
fmedian	`fmedian ksize modulation` Performs a median filter of size ksize ×ksize (ksize MUST be odd) to the original image with a modulation parameter modulation. The output pixel is computed as : out =mod × m+(1−mod)×in, where m is the median-filtered pixel value. A modulation's value of 1 will apply no modulation.
fmul	`fmul scalar` Multiplies the loaded image by the `scalar` given in argument.

FWHM *= Full-Width Half-Maximum*	The technical term Full-Width Half-Maximum `FWHM` *[https://fr.wikipedia.org/wiki/Largeur_%C3%A0_mi-hauteur]* , is used to describe a measurement of the width of a star  It is a well-defined number which can be used to compare the quality of images obtained under different observing conditions. In an astronomical image, the FWHM is measured for a selection of stars in the frame and the "seeing" or image quality is reported as the mean value. Solower value for FWHM is better. With Siril it's possible to obtain the FWHM parameters in arcseconds units. This requires you fill all fields corresponding to your camera and lens/telescope focal in the setting parameter window. If standard FITS keywords FOCALLEN, XPIXSZ, YPIXSZ, XBINNING and YBINNING are read in the FITS HDU, the PSF will also compute the image scale in arcseconds per pixel.
gauss	`gauss sigma` Performs a Gaussian filter with the given sigma.
help	`help` Gives the available commands in output logs tab.
High-speed, low-quality bilinear interpolation method	Bilinear interpolation is quite fast , color error is reduced and resolution is in-creased. Given the improvements and the modest computational cost, many devices like webcams will employ bilinear interpolation to reconstruct the image.
histo	`histo layer` Calculates the histogram of the image layer in memory and produces file histo_[layer name].dat in the working directory. layer = 0, 1 or 2 with 0=red, 1=green and 2=blue.
Histogramme	The histogram equalization increases the contrast of the image by increasing the dynamic range of the intensity given to the pixels with the most likely intensity values. It is highly recommended to evaluate all the signals contained in the image. 

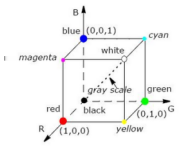
HSL *= HLS*	stands for hue, saturation, and lightness (or luminosity), and is also often called HLS • Hue • saturation • luminance 
HSV *= HSB*	The HSV, or HSB, model describes colors in terms of hue, saturation, and value (brightness). 
idiv	`idiv filename` Divides the image into memory by the image given in argument.
imul	`imul filename` Multiplies the image in memory by the image given in argument.
isub	`isub filename` Substracts the image in memory by the image given in argument.
linear	the default mode of Siril. The pixels are displayed from dark to light in a linear scale. 
Linear Fit Clipping	It fits the best straight line (y=ax+b) of the pixel stack and rejects outliers. This algorithm performs very well with large stacks and images containing sky gradients with differing spatial distributions and orientations.

load	`load filename`
	`load filename.ext`
	Loads the fits image filename; it first attempts to load filename, then filename.fit and finally filename.fits and after, all supported format, aborting if none of these are found. These scheme is applicable to every Siril command implying reading files. Fits headers MIPS-HI and MIPS-LO are read and their values given to the current viewing levels.

log	`log` Computes and applies a logarithmic scale to the current image.
Logarithm	The logarithmic scale. The operation simultaneously exacerbating the weak and light levels of the image. 
ls	`ls` This command lists files and directories in the working directory.
Median Sigma Clipping	This is the same algorithm than *`Sigma Clipping`* except than the rejected pixels are replaced by the median value of the stack.
Median Stacking	This method is mostly used for **dark/flat/offset stacking**. The median value of the pixels in the stack is computed for each pixel. As this method should only be used for dark/flat/offset stacking, it does not take into account shifts computed during registration. The increase in SNR is proportional to **0.8√N**.
Minimization	Minimization is performed with a non-linear Levenberg-Marquardt algorithm*[https://en.wikipedia.org/wiki/Levenberg-Marquardt_algorithm]* thanks to the very robust GNU Scientific Library*[http://www.csse.uwa.edu.au/programming/gsl-1.0/gsl-ref_35.html]*. As a first step, the algorithm runs a set of parameters excluding rotation angle in order to set good start values and thus, avoiding possible divergence. If σx−σy>0.01 (parameters directly computed in the 2-D Gaussian formula, see above), then another fit is run with the angle parameter. Therefore, the Siril Dynamic PSF provides accurate values for all the fitted parameters.
mirrorx	`mirrorx` Rotates the image around a vertical axis.
mirrory	`mirrory` Rotates the image around an horizontal axis.
new	`new width height nb_layers` Creates a new image filled with zeros with a size of width x height. The image is in 16-bit format, and it contains nb_layers layers, nb_layers being 1 or 3.

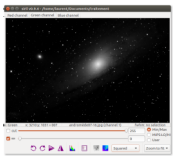
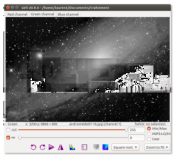
nozero	`nozero level` Replaces null values by level values. Useful before an `idiv` ou `fdiv` operation.
offset	`offset value` Adds the constant value to the current image. This constant can take a negative value. As Siril uses unsigned fit files, if the intensity of the pixel become negative its value is replaced by 0 and by 65535 (for a 16-bit file) if the pixel intensity overflows.
Patterned Pixel Grouping *= PPG*	Patterned Pixel Grouping interpolation is an advanced interpolation method which is designed to maintain the high image quality and minimize the interpolation artifacts from the output.
Percentile Clipping	This is a one step rejection algorithm ideal for small sets of data (up to 6 images).
Pixel Maximum Stacking	This algorithm is mainly used to construct long exposure star-trails images. Pixels of the image are replaced by pixels at the same coordinates if intensity is greater.
Pixel Minimum Stacking	This algorithm is mainly used for cropping sequence by removing black borders. Pixels of the image are replaced by pixels at the same coordinates if intensity is lower.
preprocess	`preprocess sequencename [-bias=, -dark=, -flat=] [-cfa] [-debayer] [-flip]` Preprocess the sequence `sequencename` using `bias`, `dark` and `flat` given in argument. It's possible to specify if images are CFA for cosmetic correction purposes with the option `-cfa` and also to demoisaic images at the end of the process with `-debayer`. The -`flip` options tells to Siril to read image from up to bottom for demosaicing operation.
Pre-processing	The preprocessing operation consists in computing the following equation: $\mathbf{calibrated\ image} = \frac{light - masterDark - masterBias}{masterFlat - masterFlatDark - masterBias}$

psf	`psf` Make a selection around a star and call the command PSF. It will give you: • The centroid coordinates (x0 and y0) in pixel units, which is the position of the center of symmetry of the fitted PSF. • The FWHM on the X and Y axis. • The rotation angle of the X axis with respect to the centroïde*[https://www.oqlf.gouv.qc.ca/ressources/bibliotheque/dictionnaires/terminologie_geomatique/centroide.html]* coordinates. • The average local background. • The maximal intensity of the star: this is the peak value of the fitted function, located at the centroid coordinates x0 and y0. • The relative magnitude of the star. • The RMSE. This is an estimate of fitting quality. The smaller the RMSE is, the better the function is fitted.
register	`register sequence [-norot] [-drizzle]` Effectue des transformations géométriques sur les images de la séquence donnée en l'argument afin qu'elles puissent être superposée à l'image de référence. Le nom de la séquence de sortie commence par le préfixe `r_`. Utilisant des étoiles pour l'alignement, cet algorithme ne fonctionne qu'avec des images de ciel profond. L'option `-norot` effectue une translation seule avec aucune création de nouvelle séquence alors que l'option `-drizzle` applique un drizzle x2 sur les images.
Registration	Registration is basically aligning the images from a sequence to be able to process them afterwards.
resample	`resample factor` Resample image with a factor *`factor`*

RGB *= RVB*	RGB (red, green, and blue) refers to a system for representing the colors to be used on a computer display. Red, green, and blue can be combined in various proportions to obtain any color in the visible spectrum. Levels of R, G, and B can each range from 0 to 100 percent of full intensity. Each level is represented by the range of decimal numbers from 0 to 255 (256 levels for each color), equivalent to the range of binary numbers from 00000000 to 11111111, or hexadecimal 00 to FF. The total number of available colors is 256 x 256 x 256, or 16,777,216 possible colors. 
rl	`rl iterations sigma` Restore an image uing the Richardson-Lucy method. `iterations` is the number of iterations to be performed (typically between 10 and 50). `sigma` is the size of the kernel to be applied.
rmgreen	`rmgreen type` rmgreen is a chromatic noise reduction filter. It removes green noise in the current image. This filter is based on PixInsight's SCNR Average Neutral algorithm and it is the same filter used by HLVG plugin in Photoshop. In command line, the lightness is always preserved. For image processing without L* preservation use the graphical tool box and uncheck the corresponding button.
rotate	`rotate degree` Rotates the image of an angle of degree value.
rotatepi	`rotatepi` Rotates the image of an angle of 180° around its center.
satu	`satu coeff` Enhance the global saturation of the image.
save	`save filename` Save current image to filename.fit. Fits headers MIPS-HI and MIPS-LO are added with values corresponding to the current viewing levels..
savebmp	`savebmp filename` Save current image under the form of a bitmap file with 8bits per channel: filename.bmp (BMP 24 bits).

savejpg	`savejpg filename [quality]` Save current image into a JPG file. You have the possibility to adjust the quality of the compression. A value 100 for quality parameter offers best fidelity while a low value increases the compression ratio. If no value is specified, it holds a value of 100.
savepng	`savepng filename [quality]` Save current image into a png file.
savepnm	`savepnm filename` Save current image under the form of a Netpbm file format with 16bits per channel. The extension of the output will be filename.ppm for RGB image and filename.pgm for gray-level image.
savetif	`savetif filename` Save current image under the form of a uncompressed TIFF file with 16bits per channel.
savetif8	`savetif8 filename` Same command than savetif but the output file is saved in 8bits per channel.
select	`select from to` This command allows easy mass selection of images in the loaded sequence (from - to, to included).
seqcrop	`seqcrop` Crops the loaded sequence.
seqfind_cosme	`seqfind_cosme sequencename cold_sigma hot_sigma` Same command than `find_cosme` but for the sequence `sequencename`.
seqfind_cosme_cfa	`seqfind_cosme_cfa sequencename cold_sigma hot_sigma` Same command than `find_cosme_cfa` but for the sequence `sequencename`.
seqpsf	`seqpsf` Same command than `PSF` but for the sequence.
Sequences	Sequences are what Siril uses to represent the manipulated files.
setcpu	`setcpu number` Defines the number of processing threads used for calculation. Can be as high as the number of virtual threads existing on the system, which is the number of CPU cores or twice this number if hyperthreading (Intel HT) is available.

setext	`setext extension` Sets the extension used and recognized by sequences. The argument "extension" can be "fit", "fts" or "fits".
setmag	`setmag magnitude` Defines the magnitude constant by selecting a star and giving the true magnitude. All PSF computations will return the true magnitude after this command.
setmagseq	`setmagseq magnitude` This command is only valid after having run seqpsf or its graphical counterpart (select the area around a star and launch the *`psf analysis`* for the sequence, it will appear in the graphs).
setmagseq	`setmagseq magnitude` This command is only valid after having run seqpsf or its graphical counterpart (select the area around a star and launch the *`psf analysis`* for the sequence, it will appear in the graphs).
Shift+Ctrl+Z	Redo
Sigma Clipping	This is an iterative algorithm which will reject pixels whose distance from median will be farthest than two given values in sigma units(σ_{low}, σ_{high}).
Simplified Drizzle x2	Activate the simplified drizzle algorithm for the processing of this sequence. An up-scale (x2) will be applied to the registered frame or during stacking depending on which registration is chosen, that will result in higher resolution images. This option is adapted for under-sampled images, i.e, when the telescope focal length is too short for the pixel size. One may consider that the system is under-sampled when FWHM is smaller than 2 pixels. The correct name of this method should be super-resolution stacking, but for a more convenient understanding we called it ***Simplified Drizzle x2***
Siril	Siril is meant to be Iris for Linux (sirI-L)
split	`split r g b` The command splits the color image into three distincts files (one for each color) and save them in r g and b file.

Squared	Square of each pixel. Which can be seen with this model viewer is primarily the most brightest part of the image. 
Square root	The square root of each pixel. This can be seen primarily in this mode are the brightest parts of the image. 
stack	`stack sequencename [type] [sigma low] [sigma high] [-nonorm, norm=]` Stack the sequence `sequencename`, using options. The allowed types are: **sum**, **max**, **min**, **med** ,**median**, and **rej** or **mean** mean that requires the use of additional arguments `sigma low` and `sigma high` used for the *`Winsorized sigma clipping`* rejection algorithm (cannot be changed from here).
stackall	`stackall [type] [sigma low] [sigma high] [-nonorm, norm=]` Opens all sequences in the CWD and stacks them with the optionally specified stacking type or with sum stacking. See STACK command for options description.
stat	`stat` Returns global statistic of the current image. If a selection is made, the command returns global statistic within the selection.
Sum Stacking	This is the simplest algorithm: each pixel in the stack is summed using 32-bit precision, and the result is normalized to 16-bit. The increase in signal-to-noise ratio (SNR) is proportional to √**N**, where **N** is the number of images. Because of the lack of normalization, this method should only be used for **planetary processing**.
threshlo, threshhi, thresh	These are threshold functions: • `threshlo 40` replaces values below 40 with 40; • `threshhi 1000` replaces values above 1000 with 1000; • `thresh 40 1000` does both.
unselect	`unselect from to` This command allows easy mass unselection of images in the loaded sequence (from - to).

unsetmag	`unsetmag` Reset the magnitude constant to 0.
unsetmagseq	Reset the magnitude calibration and reference star for the sequence.
unsharp	`unsharp sigma multi` Applies to the working image an unsharp mask with sigma sigma and coefficient multi.
Variable number of Gradients *= VNG*	This CFA recovery algorithm uses a threshold-based variable number of gradients. In order to recover missing color information at each pixel, we measure the gradient in eight directions based on a 5 X 5 neighborhood surrounding that pixel. Each gradient value is defined as a linear combination of the absolute differences of the like-colored pixels in this neighborhood. We then consider the entire set of eight gradients to determine a threshold of acceptable gradients. For all of the gradients that pass the threshold test, we use color components from the corresponding areas of the 5 X 5 neighborhoods to determine the missing color values.
visu	`visu low high` Displays an image with low and high as the low and high threshold.
wavelet	`wavelet plan_number type` Computes the wavelet transform on plan_number plans using linear (type=1) or bspline (type=2) version of the 'a trous' algorithm. The result is stored in a file as a structure containing the planes, ready for weighted reconstruction with wrecons.
Winsorized Sigma Clipping	This is very similar to Sigma Clipping method Sigma Clipping
wrecons	`wrecons c1 c2 ... cn` Reconstructs to current image from the planes previously computed with wavelet and weighted with coefficients c1, c2, ..., cn according to the number of planes used for wavelet transform

Bibliography

Adaptive Homogeneity-directed Demosaicing

Hirakawa, Keigo, and Thomas W. Parks. “Adaptive homogeneity-directed demosaicing algorithm.” Image Processing, IEEE Transactions on 14.3 (2005): 360-369.

Color filter array recovery using a threshold-based variable number of gradients

Chang, Edward; Cheung, Shiufun; Pan, Davis Y. Affiliation: AA(Compaq Computer Corp., USA), AB(Compaq Computer Corp., USA), AC(Compaq Computer Corp., USA) Publication: Proceedings of the SPIE, Volume 3650, p. 36-43 (1999).

Juan Conejero

Juan Conejero, ImageIntegration, Pixinsight Tutorial

Mighell, J. K.

Mighell, K. J. 1999, in ASP Conf. Ser., Vol. 172, Astronomical Data Analysis Software and Systems VIII, eds. D. M. Mehringer, R. L. Plante, & D. A. Roberts (San Francisco: ASP), 317.

Peter J. Huber and E. Ronchetti

Peter J. Huber and E. Ronchetti (2009), Robust Statistics, 2nd Ed., Wiley

Pic

format de Iris, ©Christian Buil

Pixel Grouping

an algorithm for Bayer color-filter array (CFA) demosaicing that developed and implemented on April 3rd, 2003 by Chuan-kai Lin*[https://sites.google.com/site/chklin/demosaic]*

SAOImage DS9

DS9*[http://ds9.si.edu/site/Home.html]* is an astronomical imaging and data visualization application. DS9 supports FITS images and binary tables, multiple frame buffers, region manipulation, and many scale algorithms and colormaps. It provides for easy communication with external analysis tasks and is highly configurable and extensible via XPA and SAMP.SAOImage DS9 is an astronomical imaging and data visualization application. DS9 supports FITS images and binary tables, multiple frame buffers, region manipulation, and many scale algorithms and colormaps. It provides for easy communication with external analysis tasks and is highly configurable and extensible via XPA and SAMP.